



# Gamification Quest: Rhythm

Music as a game mechanic



– Marina Granell Díaz –

Advisor: Dr. Raúl Montoliu Colás

Department of Engineering  
Universitat Jaume I de Castellón

Castellón de la Plana

July 2017

## *Abstract*

This document constitutes the Technical Report for the project ***Gamification Quest: Rhythm, music as a game mechanic*** for the Videogame Design and Development bachelor degree.

The project consists on the design and implementation of rhythm game mechanics integrated in a gamification environment applied to education. The video game will be implemented on the game engine Unity (10), and will constitute a playable level inside a complete gamification application. In this implemented level, the player will choose one of the available musical instruments to change the environmental music, thus generating topological and behavioral modifications on the level with the aim to overcome spatial puzzles.

## *Keywords*

*Gamification, Game Mechanics, Rhythm, Platforms, Android*



# Table of contents

---

<b>Abstract</b> .....	1
<b>Keywords</b> .....	1
Table of contents .....	3
List of figures .....	5
List of tables .....	7
Nomenclature .....	8
Chapter 1 Technical Proposal .....	9
1.1. Motivation .....	9
1.2. Project Overview .....	10
1.3. Justification .....	12
1.4. Objectives .....	14
1.5. Project Plan .....	15
1.6. Costs .....	17
1.7. Risk Management .....	18
1.8. Tools .....	19
1.9. Expected Results .....	21
Chapter 2 Game Design Document .....	22
2.1. Gameplay .....	22
2.2. Gamification .....	23
2.3. Game Mechanics .....	24
2.4. Characters .....	24
2.5. Environment .....	24
2.6. Controls .....	25

2.7. Art .....	25
2.8. Audio .....	25
Chapter 3 Game Development .....	26
3.1. General Overview .....	26
3.2. Deliverables .....	27
3.3. Game Design.....	28
3.4. Game Art .....	31
3.5. General Implementation .....	47
3.6. Artificial Intelligence Implementation.....	64
3.7. Testing .....	67
Chapter 4 Results .....	68
4.1. Objective completion .....	68
4.2. Final game state .....	69
Chapter 5 Testing & Evaluation .....	71
Chapter 6 Conclusions .....	72
6.1. Project overview .....	72
6.2. Developed competences .....	73
6.3. Future work .....	73
Chapter 7 Project Deviations .....	74
Chapter 8 References.....	77
Chapter 9 Appendix A. Gamification Quest: Rhythm GDD .....	80
Chapter 10 Appendix B. Music and Sound Design.....	81
Chapter 11 Appendix C. Playtesting questionnaire .....	82
Chapter 12 Appendix D. Project Exhibition Leaflet.....	83

# List of figures

---

Figure 1. Gantt diagram .....	17
Figure 2. Positive and negative impact of player's academic performance .....	23
Figure 3. High level representation of game relation to the gamification process .....	26
Figure 4. Tutorial layout.....	29
Figure 5. Complete level layout .....	30
Figure 6. Possible level itinerary .....	31
Figure 7. Early conceptual design of familiars .....	34
Figure 8. Blueprint of the chosen familiar .....	35
Figure 9. Process of creation of the familiar model .....	35
Figure 10. Final steps in the creation of the familiar model.....	36
Figure 11. Final familiar model with and without applied colors .....	36
Figure 12. Offensive enemy blueprint .....	37
Figure 13. Final offensive enemy model with and without applied colors .....	38
Figure 14. Bones applied to the offensive enemy 3D model.....	39
Figure 15. Passive enemy concept art .....	40
Figure 16. Passive enemy blueprint.....	40
Figure 17. Three different frames of the carnivorous plant animation .....	42
Figure 18. Familiar texture.....	43
Figure 19. Distribution of vertices along the texture in the unwrap process .....	44
Figure 20. Creating effects with the unwrap process.....	44
Figure 21. Mandolin instrument bar.....	48
Figure 22. Violin instrument bar .....	49
Figure 23. Aulos instrument bar .....	49
Figure 24. Example of tracks reproduced depending on the game state .....	50
Figure 25. Graphical variations in light caused by selecting different instruments .....	51
Figure 26. Detail of terrain topology .....	52
Figure 27. Example of terrain generated by the described method .....	53
Figure 28. Terrain with environment components.....	53
Figure 29. Collider applied to a tree .....	54
Figure 30. Generated navigation mesh for the tutorial level .....	55
Figure 31. UI with mockup elements.....	56

Figure 32. Instrument action bars.....	56
Figure 33. Pause menu.....	57
Figure 34. Pop-up messages .....	57
Figure 35. Example material using Toon Lit shader .....	58
Figure 36. Magnetic force fields .....	59
Figure 37. Three joystick positions and their effects on the avatar's movement .....	61
Figure 38. Camera position in-game.....	62
Figure 39. Negative and positive monoliths .....	63
Figure 40. Process of using a wood plank to keep open a carnivorous plant .....	63
Figure 41. Path defined by the steering behavior .....	64
Figure 42. Familiar's decision tree .....	65
Figure 43. Offensive enemy's decision tree.....	66
Figure 44. Passive enemy's movement decision tree.....	67
Figure 45. Gamification Quest: Rhythm .apk QR code .....	69

# List of tables

---

Table 1. Deliverable planning .....	15
Table 2. Game Design planning .....	15
Table 3. Game art planning.....	15
Table 4. General implementation planning .....	15
Table 5. Artificial Intelligence planning.....	16
Table 6. Testing planning .....	16
Table 7. Industry average salaries for 2013 .....	18
Table 8. Project cost breakdown by discipline .....	18
Table 9. Risks contemplated .....	19
Table 10. 3D environment elements .....	32
Table 11. Interactive objects' models .....	41
Table 12. Models with applied textures .....	44
Table 13. Particle Systems found in the game.....	59
Table 14. Familiar's behaviors descriptions.....	65
Table 15. Big impact bugs found during development.....	71
Table 16. Project plan variations .....	74



# Nomenclature

---

UI – User Interface

SFX – Sound Effect

2D – two-dimensional

3D – three-dimensional

AI – Artificial Intelligence

GDD – Game Design Document

# Chapter 1

## Technical Proposal

---

This chapter summarizes the Technical Proposal for the Video Game Design and Development Degree's final project.

### 1.1. Motivation

Gamification is nowadays a field not fully explored, but its popularity increases continuously given the amount of studies proving a positive influence on tasks in which it is used (1). On an educative environment, gamification increases students' engagement with studies (2). In spite of the term having a history, and the act of making a game of a daily activity being one commonly used, its implementation on video games is yet to be extended. Gamification Quest, a mobile gamification app, was born in order to obtain academic information from a virtual classroom and apply it to video games.

This project makes use of the Gamification Quest app to explore game mechanics. These mechanics are designed to give benefits in the video game to players who better and sooner complete the given educational tasks. The video game's graphics are 2.5D (3D objects situated on a 2D world space). The game will be implemented in Unity 3D.

As well as the gamification component, this project focuses in exploring rhythm mechanics. Rhythm as a game mechanic isn't new per se, but rhythm games are a genre with plenty of room to grow. Since the origin of music games such as *Stepmania* (3), there have been multiple branches in the genre, like musical platform games not unlike some levels in *Rayman Legends* (4). Another example would be the independent video game *Crypt of the*

## Gamification Quest: Rhythm

*Necrodancer* (5), a roguelike<sup>1</sup> dungeon exploration game, where the music is a vehicle for environment changes and composes the essentials of the player's movement.

This project proposes relating the music's mood and the sensations it awakes in the player to the environment and the way it responds to input. This way, the player has to stay focused on keeping the rhythm required to keep the instrument playing while also moving the avatar and taking decisions regarding which instrument to play and when to do it in order to solve the puzzles and keep advancing. The ability to coordinate all these actions becomes an essential one when playing the game and defies the player's capacities.

Given the gamification app's features, the video game is best understood on a mobile platform, where students can access their accounts and check their progress at any moment. The mobile platform also offers the advantage of a tactile input, beneficial for the described rhythm mechanics.

Once unlocked by completing a task, levels are always available, and the student can replay them as long as their energy level is greater than the energy required by the level. Because of this feature, the environment should support replayability, offering some kind of variation so the player doesn't get bored. The environment includes multiple starting and ending points, and the combination of them constitutes different levels or missions -but replaying a single tour is still void of purpose for a player that has already solved the puzzle. To solve this issue, the following solution is proposed –creating multiple spawn points for the object to be retrieved, creating a combination of paths for each entry and exit pair. The structure of the environment will always be the same, but the player won't be forced to repeat their previous actions when replaying a mission.

### 1.2. Project Overview

In the designed game, the player and a friendly magical creature -from now on, called familiar<sup>2</sup>- under their control are situated in a terrain full of obstacles that hinder their advance. In order to complete the level, the player has to play the given musical instruments. These instruments change the environment's looks and behavior when played, and also tell the familiar the way it has to act –such as defending the player and fetching items. Each instrument has its benefits but will also prevent the player from doing something –for example, it might eliminate an obstacle but will attract enemies.

---

<sup>1</sup> A roguelike game is one belonging to a subgenre of role-playing video games characterized by a dungeon crawl through procedurally generated game levels, turn-based gameplay, tile-based graphics, and permanent death of the player-character. (30)

<sup>2</sup> As known in European folklore, a familiar spirit that would assist witches and cunning folk in their practice of magic. (31)

## Gamification Quest: Rhythm

Thanks to the information regarding the player's educative environment, the game offers the player advantages directly related to the amount and quality of completed tasks. Because of this, players who have completed every task meticulously will have an advantage over players who fail their tasks or disregard the deliverables entirely. This difference won't ever prevent the players from completing a level. A bigger mistake buffer when playing an instrument and weaker enemies are some of the advantages won completing tasks on time.

The performance of a player in the game is also influenced by their performance in the gamified course indirectly. Given that each task completed unlocks a new mission, a player who has completed multiple tasks will have a higher level after completing the unlocked than a player who hasn't unlocked that many missions. Because the rewards' scalability, each mission rewards more experience than the previous one, and so on.

In order to reward the good work and not only the amount of it, the reward received when completing a task is directly related to its grade. This way, players will also reach a higher level and be higher in the ranking when they successfully complete a task.

Selecting an instrument affects the environment by modifying the behavior of certain elements and enemies, but also varies its appearance. Music's mood defines the lightning – its intensity and color- and some materials, completing the feeling transmitted by the instrument. The player's perception is transported from an instrument's environment to the following one in a gradual change, avoiding sudden modifications, and following the music's own transition from one instrument to another.

Each instrument follows a common melody and rhythm, and moving instruments doesn't mean playing a different melody altogether, but rather carrying from the moment the previous one left out. When incorrectly playing an instrument, an underlying deep sound increases in volume at the same time the music momentarily lowers, generating a sensation of wrongness in the player and alerting them of something not going like it should. Varying the appearance of the game along with the instrument directly relates the mechanic with a response and generates a satisfying feeling for the player, whose every action is rewarded with a consequence.

Player movement will be directed by a virtual joystick. Every other element will be controlled by touches on custom screen elements. All in all, there is a button for each instrument that let the player change the selected instrument, and an action bar for the selected one. This bar is meant to provide a mean for the player to play the instrument. How each instrument is played varies, but the bar will always maintain a vertical design and a moving mark along its height. This mark will move in time with the music.

## Gamification Quest: Rhythm

The player has to constantly play the selected instrument in order to keep the familiar alive. If the fail limit is exceeded, then the creature dies and the player loses. Successfully playing the instrument restores the previous fails and recovers the creature.

There are four kinds of elements that react to music in the environment: passive enemies, offensive enemies, carnivorous plants and magnetic monoliths. Each one will react to one or more instruments, producing varying effects. In the following list, some of the effects are detailed:

- Passive enemies: they move towards the player when an instrument that attracts them is played. They block the path and only attack when the player comes too close to them.
- Offensive enemies: they follow the player and attack unless an instrument that repulses them is playing. The familiar can defend the player against them as long as the instrument they're playing allows it.
- Carnivorous plants: they will open if the instrument they like is being played, allowing the player to go over them. They will close as soon as the player stops playing that instrument. They can be kept open by placing an object in their insides.
- Magnetic monoliths: they react to an instrument by reversing their polarity from positive to negative and vice versa. The player can activate their own polarity, always set positive, by playing a different instrument. The activation of both polarities results in the player being attracted or repelled by the monoliths. The force produced enables the player to cross bodies of water.

The familiar will also react to the instruments changing its behavior. There are a total of three different behaviors: defend, interact and retrieve. While defend allows the creature to protect the player against offensive creatures, the other behaviors are related to the interaction with the environment. Interact makes the creature use elements of its surroundings up until a certain distance, called range. Retrieve makes the creature pick up objects inside its range. Activating interact mode while holding an object will result in the use of this object on the creature's surroundings if the possibility exists.

### 1.3. Justification

This section presents the relation of the following courses included in the bachelor's degree and this project. Some of the courses inspired the project altogether, and the contents of others made developing it possible. While many of the courses in the degree are related to the project, only those further related are discussed.

## Gamification Quest: Rhythm

### 1.3.1. Conceptual Video Game Design (VJ1222)

The study of game mechanics, the relation between the player and the game and the existent genres included in this course inspired the creation of the mechanics applied in this project. The design of environment related to the mechanics is one of the basic skills developed in this course. Correctly accomplishing the design and describing it is also a key factor in the development of this project. Up next, the key competences regarding the course are listed.

**E15** – Define rules that harmonize with technical possibilities and contribute to fluidity, coupled with game mechanics as the main components in the creation of the design.

**E15, G10** – Design balanced sceneries and environments at the service of the game development.

**E15, G10** – Design relations between game mechanics, objectives to achieve and the possible setbacks in and out of the game.

**E15, G10, G05** – Develop the information regarding game design in a clear, precise and orderly fashion.

**G04** – Write a document regarding video game design in technical English.

### 1.3.2. 3D Design (VJ1216) & Character Design and Animation (VJ1226)

Designing and modelling the environment is essential in the creation of the proposed game. Creating objects artistically and with a low polygon count for performance is a skill developed in these courses, as well as the use of the needed software. The process of animating and exporting these objects and characters is also detailed in them. Up next, the key competences regarding the courses are listed.

**E04** – Ability to design and create graphical elements.

**E04** – Use of tridimensional modelling applications for video games.

**E06** – Ability to design and create animated characters.

**E06, IR07, IR13** – Use of character modelling and animation applications for videogames.

### 1.3.3. Artificial Intelligence (VJ1231)

Designing the behavior of computer controlled characters and elements enables the player to interact in a varying fashion with an environment that would otherwise turn boring easily.

## Gamification Quest: Rhythm

The conceived behaviors should be implemented by choosing the correct algorithms. Up next, the key competences regarding the course are listed.

**IR06** – Knowledge and application of basic algorithmic procedures of computing technologies for the design of solutions to problems, analyzing the suitability and complexity of the proposed algorithms.

**IR08** – Ability to analyze, design, build and keep video games in a stable, secure and efficient way, choosing the most suitable paradigms and the programming languages.

**IR15** – Knowledge and application of essential principles and basic intelligent system techniques and their implementation.

**IR06, IR08, IR15** – Solve video game problems using artificial intelligence techniques.

### 1.3.4. Sound Production and Realization Techniques (VJ1236)

The concept of the game proposed is born from the knowledge that sound is often left aside in the design of video games, and the multiple ways its consideration in the earlier design phases produces a more complete result. Up next, the key competences required to include sound as one of the main assets of the game are listed.

**E10** – Ability to design and create sounds and soundscapes.

**E14** – Ability to create production developments in multiple formats, mainly on the multimedia and video game fields, as well as on interactive television.

**E10, E14** – Develop autonomously audio production projects for videogames.

## 1.4. Objectives

This project is meant to experiment with rhythm game mechanics, as well as gamification in video games. Ideally, the finished product would provide a completely playable game in the context of the gamification app. Nevertheless, the objectives for the project must be quantifiable and measurable. Taking into account the project's features and these limitations, the following list summarizes the objectives:

1. Implement game mechanics that reward students' actions, taking advantage of the gamification application in which the game is integrated.
2. Integrate music as a game mechanic on the environment.
3. Design replayable levels without modifying their structure.

Given these objectives, the tasks in which the project is divided should prioritize playability and design over the completion of the game.

# Gamification Quest: Rhythm

## 1.5. Project Plan

In Table 1, Table 2, Table 3, Table 4, and Table 5 tasks are divided by category. As it will be discussed in Section 5.6., tasks have been sorted based on their importance for the final game. Each task has been assigned an ID in order to easily identify them.

Table 1. Deliverable planning

ID	Task Name	Start Date	Dependences	Duration (h)
D1	Technical Proposal	30 Jan		10
D2	Game Design Document	20 Mar	GD1	18
D3	Technical Report	18 May	Project in advanced state	25
D4	Presentation Video	8 Jul	Project complete	1
D5	Final Presentation	6 Jul	Project complete	5
			<b>TOTAL HOURS</b>	59

Table 2. Game Design planning

ID	Task Name	Start Date	Dependences	Duration (h)
GD1	Game mechanics	2 Feb		15
GD2	Level design with multiple entry and exit points	9 Feb	GD1	3
			<b>TOTAL HOURS</b>	18

Table 3. Game art planning

ID	Task Name	Start Date	Dependences	Duration (h)
A1	Environment	13 Feb		20
A21	Familiar modelling	20 Feb		5
A22	Familiar animation	23 May	A6	5
A31	Offensive enemy modelling	21 Feb		5
A32	Offensive enemy animation	25 May	A6	5
A41	Passive enemy modelling	15 May		5
A42	Passive enemy animation	27 May	A6	5
A51	Interactive objects modelling	17 May		5
A52	Interactive objects animation	29 May		5
A6	Texturizing	29 Mar	A1,A21,A31,A41,A51	5
			<b>TOTAL HOURS</b>	65

Table 4. General implementation planning

ID	Task Name	Start Date	Dependences	Duration (h)
G1	Mandolin action bar	23 Feb	GD1	7
G2	Violin action bar	25 Feb	GD1	7



## Gamification Quest: Rhythm

<b>G3</b>	Aulos action bar	28 Feb	GD1	7
<b>G4</b>	Instrument change	2 Mar	GD1,G1,G2,G3	5
<b>G41</b>	Graphical variations	10 Apr	Complete graphics	15
<b>G5</b>	Terrain creation	1 May	GD2	5
<b>G6</b>	Environment setup	3 May	G5	8
<b>G7</b>	User interface	13 Mar	GD1	4
<b>G8</b>	Input setup	15 Mar	GD1	2
<b>G9</b>	Level graphical appearance	1 Jun	G6	15
<b>G10</b>	Player & Camera movement	16 Mar		5
<b>G11</b>	Interactive objects response	8 May	G4	15
<b>TOTAL HOURS</b>				95

Table 5. Artificial Intelligence planning

ID	Task Name	Start Date	Dependences	Duration (h)
<b>AI1</b>	Familiar AI	4 Mar		25
<b>AI2</b>	Offensive enemy AI	3 Apr		10
<b>AI3</b>	Passive enemy AI	19 May		10
<b>TOTAL HOURS</b>				45

Table 6. Testing planning

ID	Task Name	Start Date	Dependences	Duration (h)
<b>T1</b>	Feature testing	23 Feb	After each new implementation	5
<b>T2</b>	Complete game testing		Complete game	3
<b>T3</b>	Post-Testing modifications		T2	10
<b>TOTAL HOURS</b>				18

The Gantt diagram shown in Figure 1 represents task duration in a visual manner. The diagram was created thanks to a Power Point extension called Office Timeline (6). Task duration is represented by dividing its hours in three to four hour work days.

# Gamification Quest: Rhythm



Figure 1. Gantt diagram

## 1.6. Costs

To complete this project, 300 working hours divided along 5 months with working days from 2 to 5 hours were spent along three different disciplines –programming, design, art and production. The time and material cost required for the completion of the project are listed in this section in order to obtain an estimate budget.

## Gamification Quest: Rhythm

Table 7. Industry average salaries for 2013

Role	Yearly income (€)* <sup>1</sup>	Hourly income (€/h)* <sup>2</sup>
<b>Junior Videogame Programmer</b>	39.398,91	20,52
<b>Junior Artist and Animator</b>	36.580,84	19,05
<b>Junior Game Designer</b>	39.115,60	20,37
<b>Junior Producer</b>	48.595,76	25,31

\*<sup>1</sup> Average yearly income in Europe, in year 2013 (7).

\*<sup>2</sup> Yearly income divided in 8 hour work days.

Table 8. Project cost breakdown by discipline

Task	Role	Hours	Salary/€	Total salary
<b>Design</b>	Junior Game Designer	18	20,37	366,66
<b>Programming</b>	Junior Videogame Programmer	158	20,52	3.242,16
<b>Art</b>	Junior Artist and Animator	65	19,05	1.238,25
<b>Production</b>	Junior Producer	59	25,31	1.493,29
			<b>Total:</b>	6.340,36

Extra costs such as the mobile device required and hardware have not been quantified, as the actual acquisition wasn't required thanks to previous projects.

### 1.7. Risk Management

Problems can arise during the development, causing delays and even preventing the project from being finished on time. To reduce the probability of this happening, Table 9 lists all contemplated risks and the course of action that can prevent their apparition or mitigate their effects.

# Gamification Quest: Rhythm

Table 9. Risks contemplated

ID	Name	Probability	Impact	Effect	Mitigation
R1	Incorrectly assessing task durations	---	--	Not all tasks are completed by the end of the project	Start working on the most important tasks as soon as possible
R2	Finding a disabling bug during testing	-	---	Game can't be played	Meticulously test after each task completion
R3	Project data loss	-	---	Can't make the delivery deadline	Create data backups
R4	Not getting the music for each instrument on time	--	--	Game lacks one of its main features	Prepare placeholders for each track
R5	Lacking skills required to complete a task	---	--	A task could be left unfinished	Learn skills required
R6	Game is too difficult to play because of lacking coordination	-	--	Players get frustrated and won't play	Modify game design within the realms of possibility.

## 1.8. Tools

This section details tools used in the creation of the described project. While most of the software requires a working license, Krita (8) and Audacity (9) are open source and provide completely working solutions for some of the tasks described in Section 1.7. Open source software has always been prioritized over licensed software – which has been used only when in need of otherwise missing functions.

### 1.8.1. Development

The described game is developed in Unity 3D (10), a generic and free game engine that supports every requirement the project could depend on. Unity 3D supports both 2D and 3D elements, an essential feature for the creation of interactive UI. Unity supports coding in JavaScript (11), C# (12) and Boo (13). However, most of the documentation and official

## Gamification Quest: Rhythm

references for Unity are written in C#, and this programming language is also one of the most used during the bachelor's degree. Because of this, all scripts are written in C# for this project.

### *1.8.2. Art and Graphics*

The game is composed by 2D and 3D elements. For this reason, specific software is needed to create both UI elements and game objects.

In order to create buttons, sliders, icons and other UI elements, both Adobe Photoshop (14) and open source Krita (8) are suitable options. While Krita offers high customization for brushes, which makes easier the creation of textures, Photoshop has a more stable support and a wider variety of filters for the creation of UI elements. Both programs are used during different design phases.

The chosen software for the creation of 3D elements and their animation is Autodesk 3Ds Max (15). This software supports polygonal modelling, ideal for the control of the total poly count of each model. Furthermore, it incorporates rigging and skinning tools, as well as a full animation support. Models are then easily exported for their use in the selected game engine.

### *1.8.3. Sound and SFXs*

Sound edition is essential in the creation of every game. Even in the most basic game, the sound produced by the click of a button leads the user to think their action was correctly received. To modify and create sounds, audio edition software is a necessary tool.

The required software should be able to extend, combine and cut audio files, modify its equalization and apply filters, but composing sounds from scratch isn't necessary. For these reasons, open source Audacity (9) fits the criteria.

### *1.8.4. Document edition*

In Section 1.5, the required documents were listed -text documents and presentations among them. In the creation of these documents, the text editor Microsoft Office Word (16) was essential. The elaboration of the first Technical Proposal used a different text editor –Overleaf (17), an online LaTeX and Rich Text writing tool. However, the missing knowledge required to further modify templates in LaTeX weighted in the decision to migrate document creation to Word.

To create the final presentation, Microsoft Office Power Point (18) offers key functionalities, such as templates, transitions, popping elements and annotations.

# Gamification Quest: Rhythm

## 1.8.5. Video edition

In order to correctly showcase definite functionalities and create a short trailer for the described game, screen recording and basic video edition are required. Camtasia Studio (19) provides solutions for easily recording audio and screen, and incorporates a powerful video rendering tool.

## 1.9. Expected Results

By the end of the project, it's expected for a game level to be completely implemented, ready for its incorporation in the gamification application. The level should contain multiple entry and exit points, as well as different objectives, offering then multiple paths for increased playability. Fluid interactions and easy implementation of new levels are to be expected.

The main concern is for the executable to be able to showcase the designed mechanics and controls. The playability must include start and end points, the objective that must be retrieved and elements that contribute to make necessary the use of the available controls.

# Chapter 2

## Game Design Document

---

This chapter discusses game design decisions and details. The complete design of the game developed has been described minutely on Appendix A. However, the main information regarding the most important elements in the game is summarized in Sections 2.1 to 2.7.

### 2.1. Gameplay

The player unlocks new levels in the game thanks to completing tasks in the virtual classroom. Once the player starts the level, they find themselves at the start point randomly selected by the game depending on the user's performance on the gamification app. The player must then find an objective marked on the map, retrieve it, and make it to the exit point. While the game design contemplates the existence of multiple levels, this project has seen to completion a tutorial and a single complete level.

In order to navigate the level, the player must play one of the three available instruments. Each instrument causes different effects on the environment and the objects and enemies in it, and directs the actions of the familiar that follows the player. While controlling the avatar's movement and playing the instrument, the player has to decide the correct course of action which will help them complete the level, evading the enemies and obstacles.

The player must not stop playing the selected instrument, being that each miss or fail counted lowers the familiar's energy and reaching the lowest point automatically ends the level. The familiar can also lose energy by receiving damage imparted by enemies.

# Gamification Quest: Rhythm

## 2.2. Gamification

The game described in this section is part of a gamification application which has been separately developed. This application connects to a virtual classroom using the player's information, and applies the information received to connect the experience in class to the game. In order to unlock new levels, the player must deliver tasks to the virtual classroom.

The app will reward students by giving them *World Keys*, the object used to unlock missions, as well as experience. Each mission gives the player different rewards and objects depending on its difficulty. When the player first enters the game, only the provided tutorial is already unlocked. Every mission except the tutorial takes the user to a different level, with multiple itineraries, so replaying a mission doesn't provide the same game experience. Every mission can be completed multiple times. For this project, only the tutorial and the first level have been implemented, meaning there are two playable levels, one of them with multiple entry and exit points as well as different objectives.

The game rewards –or punishes- the player for their work or lack of by adapting the strength of the enemies and difficulty of the itinerary inside the level (from the entry point to the objective, and to the exit) to the time the user usually takes to deliver a task in the virtual classroom (from 1 to 0, where 1 is delivering it just as it opens and 0 is delivering at the last minute) and the obtained grades. Figure 2 summarizes the academic information's impact in the game.

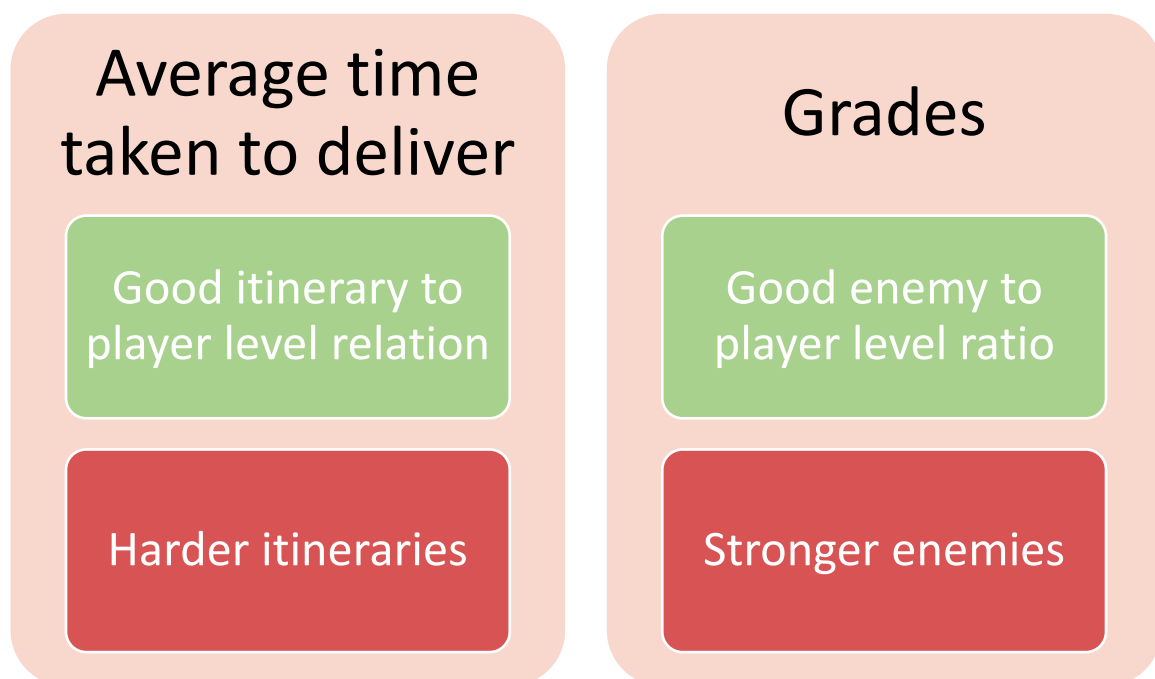


Figure 2. Positive and negative impact of player's academic performance



## Gamification Quest: Rhythm

Itinerary difficulty is decided based on the number of instrument changes necessary to complete the level and the kind of situations the player will find while completing it. For example, an itinerary that doesn't include magnetic monoliths would be easier than one that does.

### 2.3. Game Mechanics

The player must move and play an instrument simultaneously in order to advance to their objective. There are three instruments, played differently, that produce different effects on the game. These are the mandolin, the violin, and the aulos. Selecting a new instrument changes the music and the light of the environment, and causes changes in the object's behavior.

Effects are explained when the object affected is introduced. The only effect regarding solely the player is the activation of the avatar's magnetism, by selecting the aulos.

### 2.4. Characters

There is an only playable character. This character, called avatar, is the vehicle for the player to interact with the game.

The player can, however, select the familiar that follows the avatar from the list of unlocked familiars. The familiar is a Not Playable Character (NPC). Familiars can be obtained as a reward to some levels. The familiar can behave differently depending on the selected instrument. There are three possible behaviors: retrieve, interact and defend. When playing the mandolin, the familiar enters retrieve mode and picks up object in its range. Playing the violin will activate interact mode. If the familiar can interact with an object in its range, it will do so. Lastly, playing the aulos activates defend mode. When defending, the familiar intercepts and returns attacks thrown by enemy NPCs.

Enemy NPCs can be found on the levels. There are two kinds of enemies: offensive, and passive. Passive enemies only follow the avatar when playing a certain instrument, and attack when the player approaches too much. Offensive enemies, though, always attack the avatar when they enter their action range.

### 2.5. Environment

Environment objects can be classified in two categories: interactive, and decorative. Interactive objects react to the player and the selected instrument, while decorative objects are only meant to complete the aesthetic of the scenery.

## Gamification Quest: Rhythm

Some interactive objects react to the selected instrument and change their behavior. This is the case of the carnivorous plants and the magnetic monoliths. Carnivorous plants selectively let the player through depending on the selected instrument (only when playing the mandolin, the player can go past them). Magnetic monoliths change polarity each time the player selects the violin, from positive to negative and vice versa. Monoliths attract or repel the avatar when the player activates its polarity.

There are interactive objects meant to be picked up by the familiar using retrieve mode. Wood planks are picked up by the familiar and, when used to interact with an open carnivorous plant, block the plant's jaws keeping them open.

### 2.6. Controls

The game is controlled by tactile input received by the mobile device screen. The game uses a horizontal format to better distribute interface elements meant to control the game. The player controls the avatar's movement by moving a virtual joystick situated on the left bottom side of the screen. On the right side of the screen, the virtual device meant to control the instruments is placed.

### 2.7. Art

The game contains both 3D and 2D art. 2D elements are exclusively used on the interface, with the exception of warning signs and question marks placed over 3D elements in the environment, meant to guide the player.

3D elements use flat shading and saturated colors to bring up contrast between different elements. These 3D assets use a low-poly aesthetic in order to keep low the total polygon count and relieve the mobile device from unnecessary rendering processing time.

### 2.8. Audio

The game includes both music and sound effects (SFXs). Each instrument produces a different melody, but all of them contains the same base and is similar to the rest. Sound effects are only used as auditory feedback. The sound design was left in the hands of the composer Víctor Ávila, from the *Conservatori Superior de Música de Castelló* (20). This collaboration started thanks to Miguel Chover, Director of the Game Design and Development Degree, who established contact between both parties. Details can be found in Appendix B.

# Chapter 3

## Game Development

---

### 3.1. General Overview

As described in Chapter 1, the game has been designed with the objective of its integration into a gamification application and makes use of the information it gathers about the player. Figure 3 shows a high level representation of how the game is connected to the rest of the application and what use the received information serves.

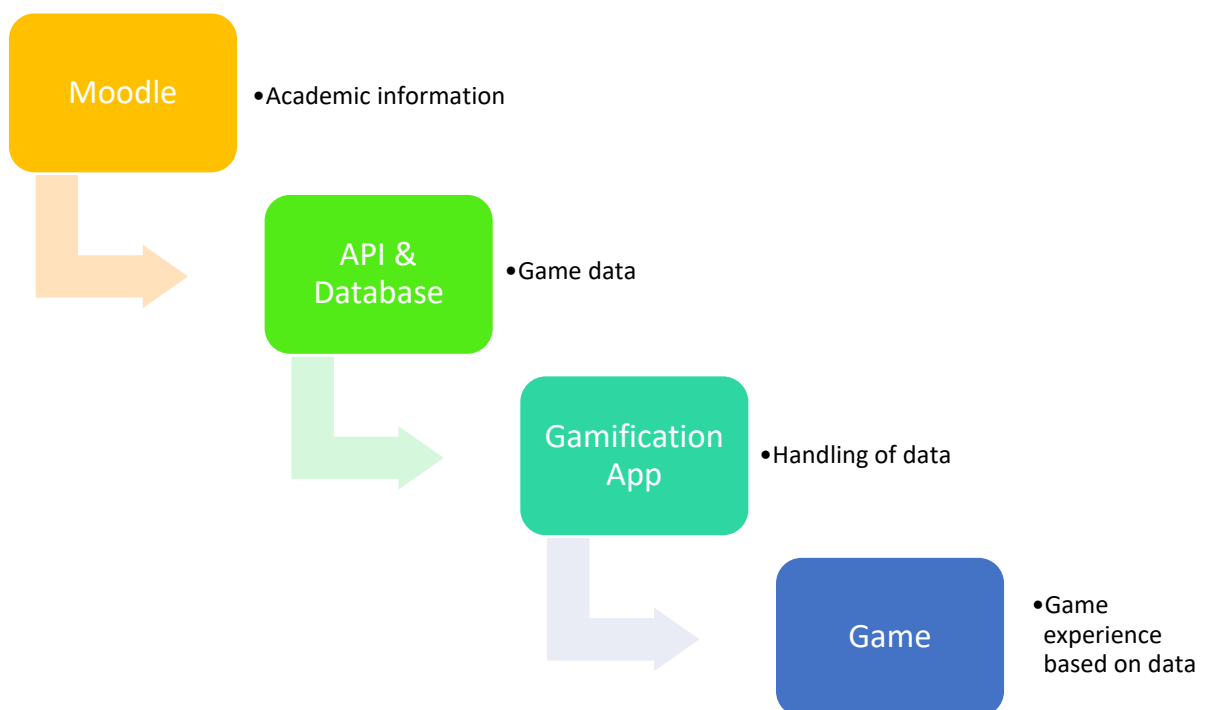


Figure 3. High level representation of game relation to the gamification process

# Gamification Quest: Rhythm

## 3.2. Deliverables

This section summarizes the content of every deliverable and the tools used to produce them.

### *3.2.1. D1. Technical Proposal*

The Technical Proposal was redacted using Overleaf (17). This proposal was redacted starting on Raúl Montoliu's course. The final proposal was product of several tutoring sessions and revisions. This proposal was the base for the Technical Proposal included in Chapter 1, and described the intended work for this project.

### *3.2.2. D2. Game Design Document*

The Game Design Document (GDD) has been included in this document as Appendix A. A concise version of the information provided in the GDD composes Chapter 2. This document was created using Microsoft Word (16). The GDD provides the information regarding game mechanics design, graphical elements, behavior of specific elements, audio, and other decisions regarding design.

### *3.2.3. D3. Technical Report*

The Technical Report has been redacted using Microsoft Word (16). This document includes all the information regarding the development of the project, including extended versions of the previous Technical Proposal. The aim of this document is to describe the development of the project and document it.

### *3.2.4. D4. Presentation Video*

The presentation video has not been produced at this stage of the project. The software meant to be used to produce it is Camtasia Studio (19). In order to record the phone's screen, the AZ Screen Recorder (21) app is meant to be used. This video is meant to showcase the game's main functionalities while remarking the project's main objectives.

Not included in the deliverables section, as it isn't officially part of the evaluation process, is the project exhibition. This video will be showcased in this project's presentation at the exhibition. A leaflet included in Appendix D has been designed in order to simplify the downloading process for the attendance. The leaflet includes a QR code that, when scanned, directs the mobile device to the download and installation of the videogame. This way, attendants will be able to test the game in their devices easily. This leaflet also includes basic information about the game.

## Gamification Quest: Rhythm

### *3.2.5. D5. Final Presentation*

To display the information during the presentation of the project, slides created using Microsoft PowerPoint (18) and containing graphics, screenshots, videos and supporting text.

## 3.3. Game Design

### *3.3.1. GD1. Game mechanics*

Game mechanics are an essential part of the project. If the mechanics were flawed, the project as a whole wouldn't make sense. Mechanics must relate the academic work of the player with difficulty and rewards in order to take advantage of the information provided by the gamification system described in Appendix A, as well as provide the player with an entertaining experience.

The game mechanics are extensively explained in Appendix A.

### *3.3.2. GD2. Level design*

As stated in the project's objectives in Section 1.4, a level must provide multiple itineraries to increase replayability. This means there must be multiple possible positions for the start point, the objective and the exit point. The combination of these different positions results in the creation of different itineraries. Nevertheless, in order to teach the player how to play, a tutorial level that includes some of the basic elements has been designed. This level is independent from the rest of itineraries.

Figure 4 shows the tutorial level and its elements. This level is meant to introduce the player to the mechanics of the game –retrieving an object, and going to the exit point- as well as introduce them to the coordination between movement and playing the chosen instrument.

## Gamification Quest: Rhythm

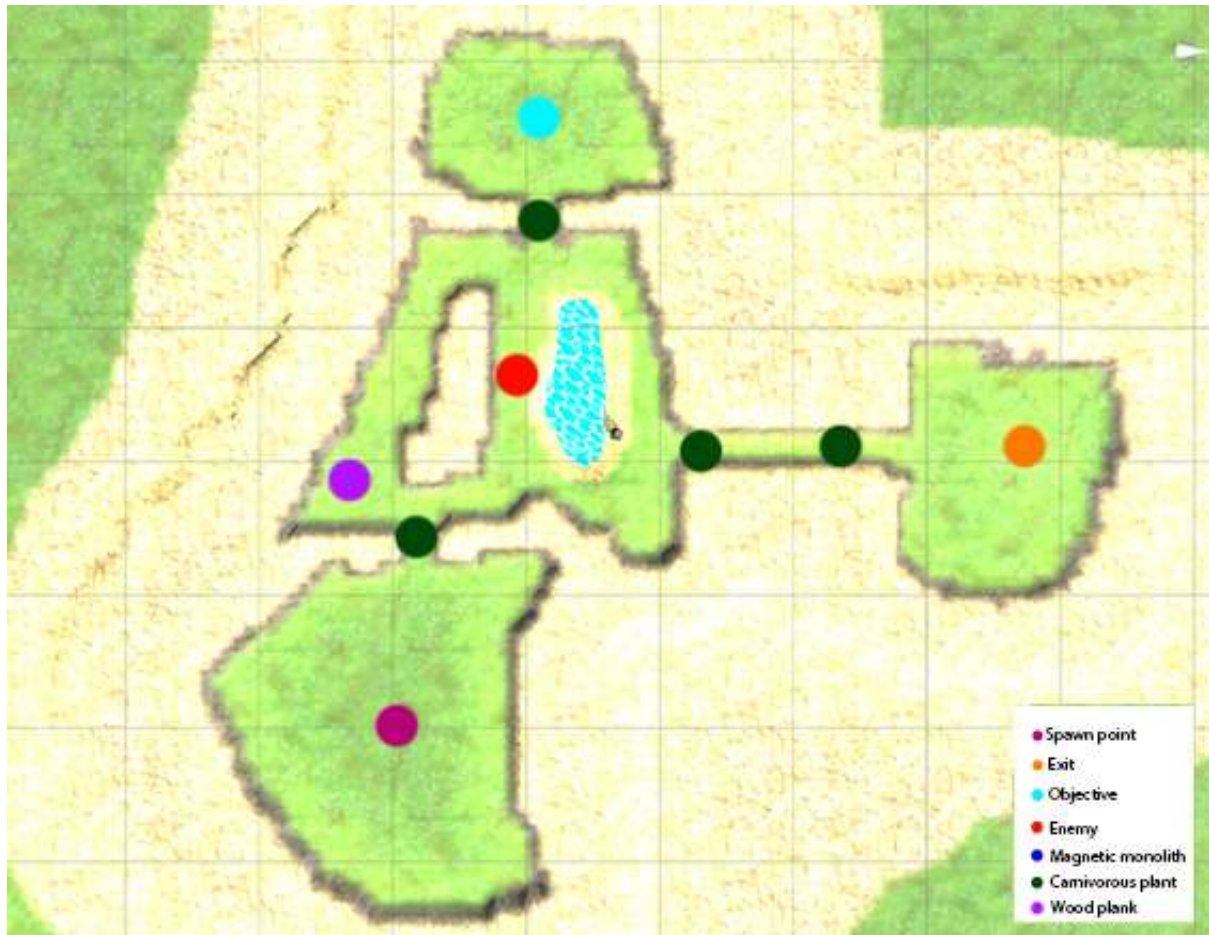


Figure 4. Tutorial layout

The full level includes a total of seven player spawn points, seven different objectives and five exit points. The entry, objective and exit are randomly selected each time the player selects a level, while keeping a minimum distance between each pair of positions to keep the player from being spawned right next to their objective or the exit. Figure 5 shows the complete level layout and its elements. This level portrays the complete extent of possibilities of the applied game mechanics.



## Gamification Quest: Rhythm

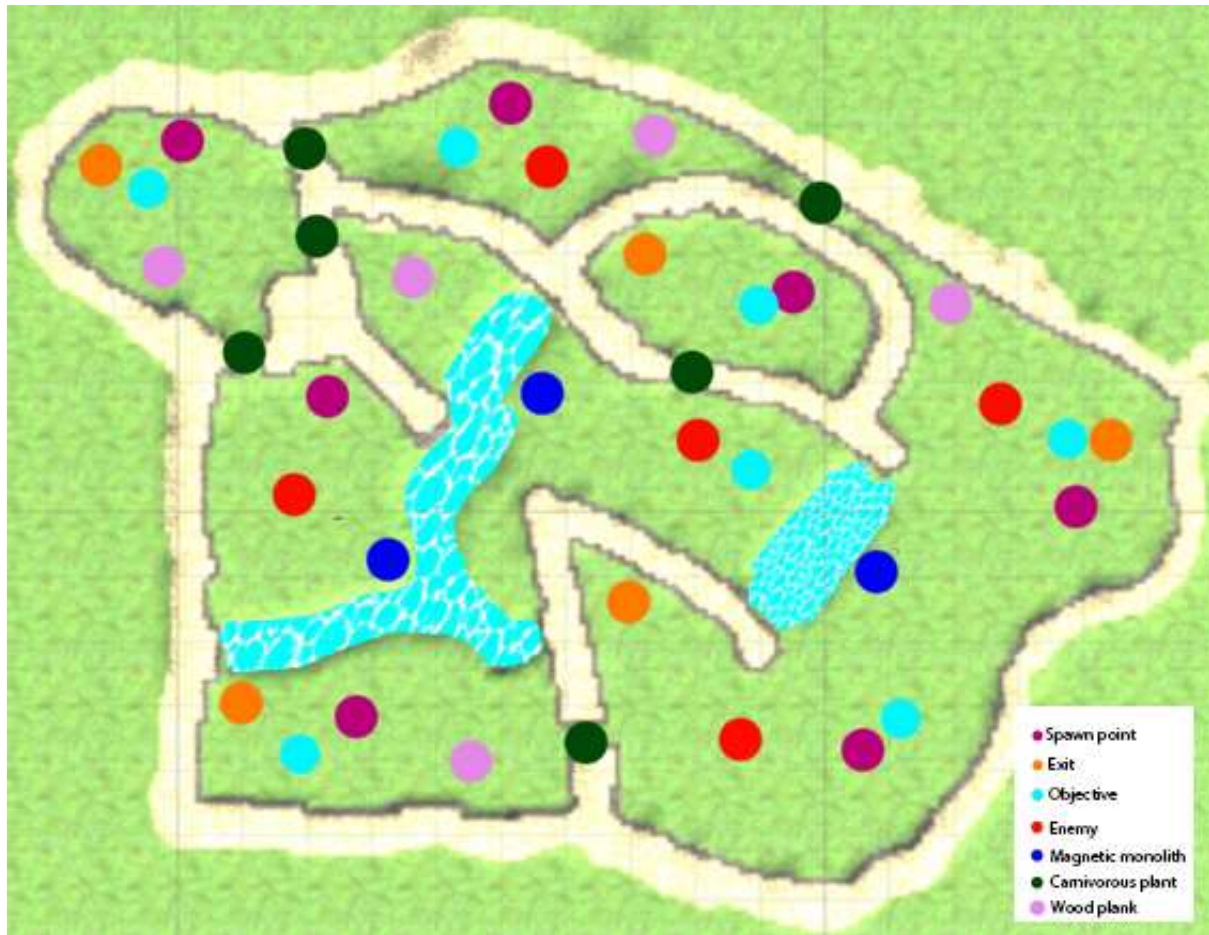


Figure 5. Complete level layout

Any combination between entry point, objective and exit is possible as long as they are kept apart from each other. In Figure 6 a possible itinerary is marked. Zones where interactions between the player and other game elements might happen are marked.

## Gamification Quest: Rhythm

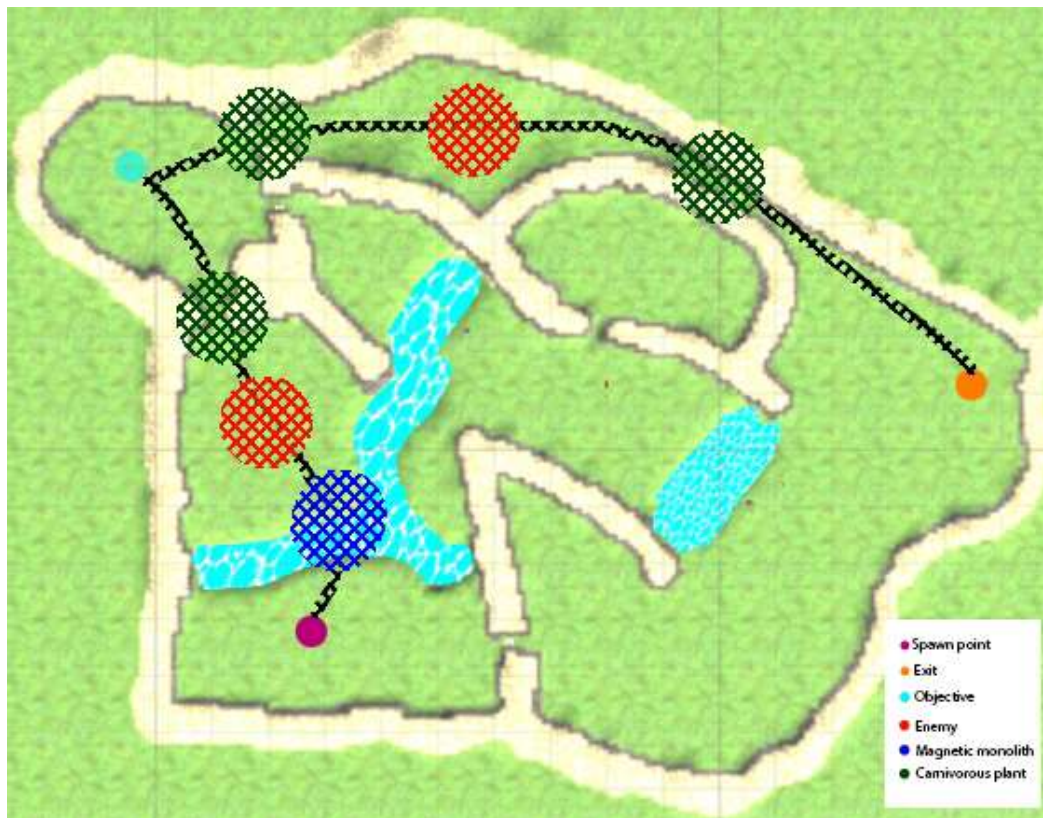


Figure 6. Possible level itinerary

### 3.4. Game Art

This section describes the creation of all the models used in the game. All the 3D elements have been specially modelled for this project. The only exception is the definitive avatar, which has been modelled by Luis Alisandra, as he also uses the gamification app in his project. Sharing assets is meant to unify the app's aesthetics.

#### 3.4.1. A1. Environment

The design of the environment and its components followed the process described in this section.

First, a conceptual design regarding the game mechanics hinted that the environment should be an open space, where the theme of magic and ethereal creatures wouldn't be out of place. Thus, the decision to recreate a forest was made.

In order to keep the application's size small and minimize rendering time, being that mobile platforms vary in storage space and processor capacity, 3D models should keep a low polygon count. This is the reason the game acquired a low-poly aesthetic.

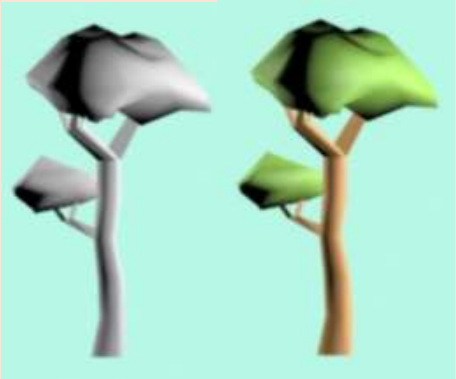
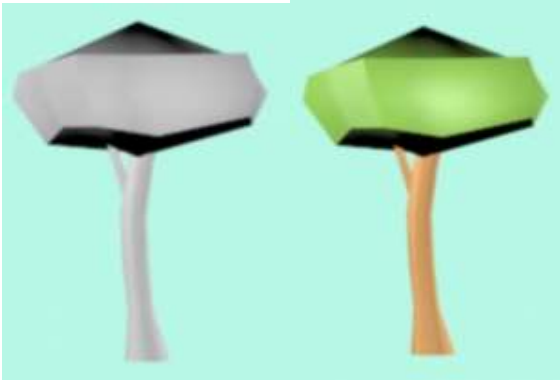


# Gamification Quest: Rhythm

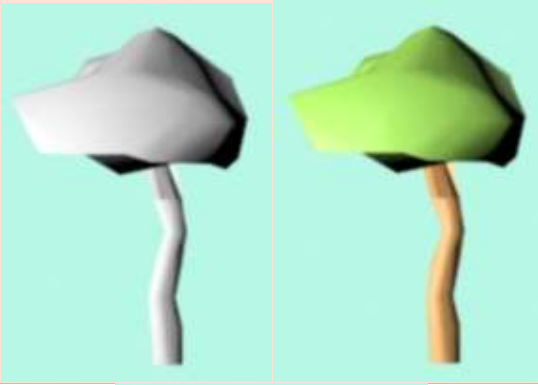

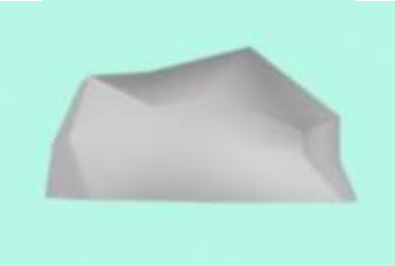
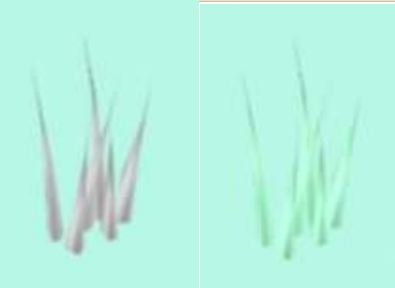

Environment components should include obstacles, elevations and depressions that make the level interesting from a playability stand. Decorative elements are also necessary in order to keep open spaces from looking bare. These elements shouldn't impede the player from moving through them –unlike obstacles, with which the player should collide if attempting to go through them.

While obstacles and decorative elements could be modelled using 3DS Max (15), elevations and depressions are better understood when in context of the whole terrain. As such, their creation will be detailed in Section 3.5.6. In **¡Error! No se encuentra el origen de la referencia.**, 3D elements are listed by name. Models are shown both without and with flat colors. Textures will be applied at a later stage.

Table 10. 3D environment elements

Name	Render	Polygon Count
Tree I		320
Tree II		152

## Gamification Quest: Rhythm

Tree III		302
Rock I		24
Rock II		32
Grass		72
Flowers		315

Every model has been created based on primitives such as cylinders, boxes and spheres. Flower petals are an exception, as they were modelled from planes and are two-dimensional in order to keep polygon count as low as possible.

## Gamification Quest: Rhythm

### 3.4.2. A21. Familiar modelling

Familiars are ethereal creatures, spirits that follow the player. While they preserve animal qualities, they are not animals per se. For this reason, conceptual designs shown in Figure 7 aren't based in a single animal and are not existing mythical creatures.

After selecting a definite design, it is necessary to create a blueprint that contains all the important representations –usually, front and lateral representations- as well as defined the creature's proportions. This blueprint, shown in Figure 8, serves as a base for the creation of the 3D model.

The next step in the process is the creation of the 3D model. After inserting the blueprints in the 3D environment to use as a reference, each part is modelled parting from primitives. Figure 9 and Figure 10 show key steps in the process of modelling.



Figure 7. Early conceptual design of familiars

## Gamification Quest: Rhythm

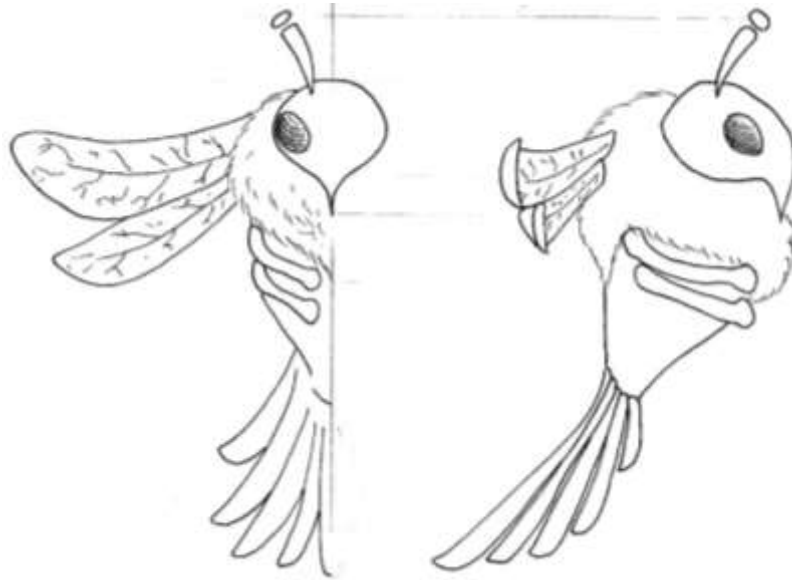


Figure 8. Blueprint of the chosen familiar

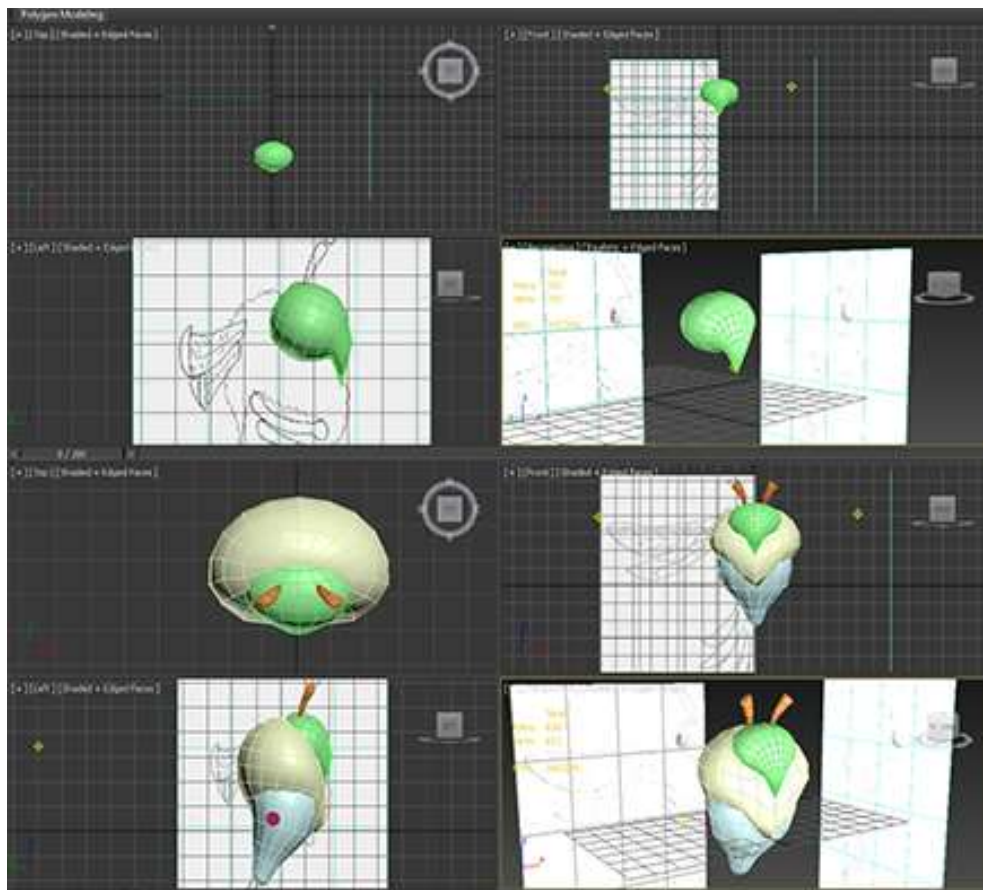


Figure 9. Process of creation of the familiar model

## Gamification Quest: Rhythm

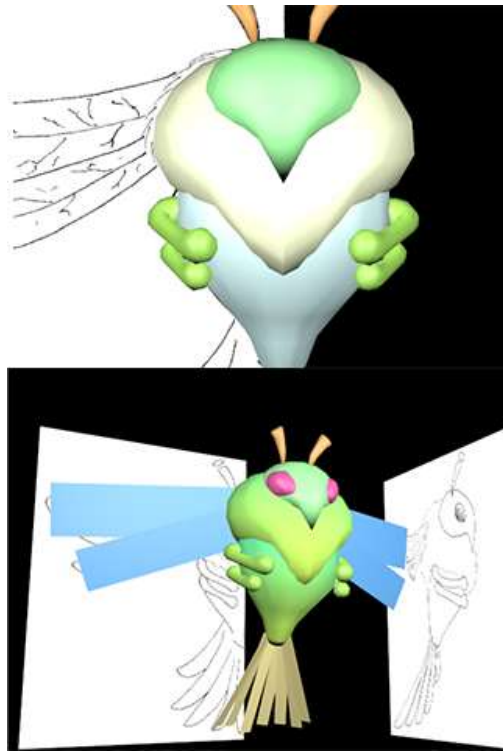


Figure 10. Final steps in the creation of the familiar model

As shown in Figure 11, the final model contains **1510 polys**. This low-poly model is ideal for its use in a mobile platform thanks to its low polygon count, while the appearance of the original design has been fully preserved.

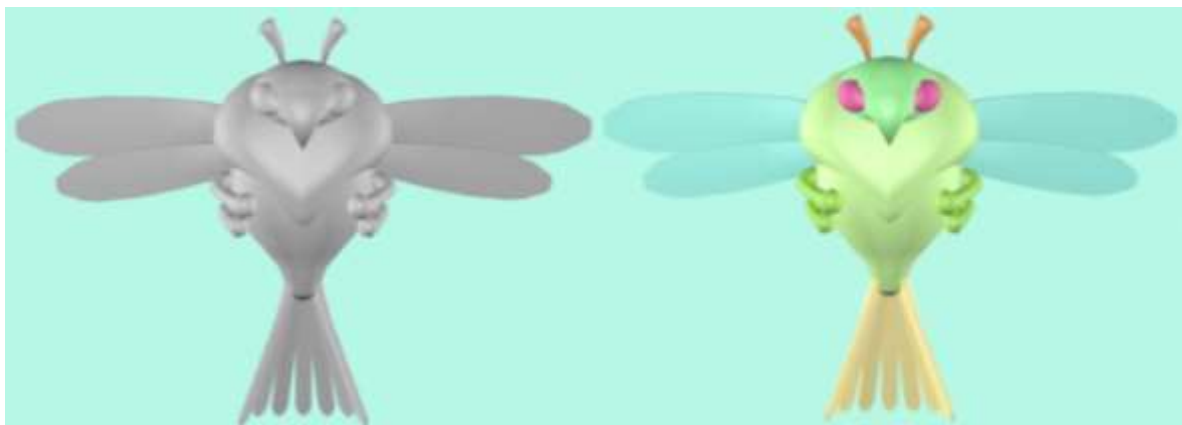


Figure 11. Final familiar model with and without applied colors

The texturizing of this model will be discussed in Section 3.4.10.

### 3.4.3. A22. Familiar animation

The familiar animation was finally replaced by movement by script, as described in Chapter 7. The scripts used rotate the familiar model and move it to recreate the illusion of animation.

## Gamification Quest: Rhythm

Another script simulates a levitating movement, moving the model up and down softly to imitate hovering.

### 3.4.4. A31. Offensive enemy modelling

The process of creation of this model was not unlike the previous ones. After selecting its design and drawing its blueprint –shown in Figure 12, the model was created by using primitives and modifying them. The body of this particular model was created using a sphere for its head and extruding its connection to the neck. Each section has been scaled according to the thickness and shape of the body in the corresponding section.

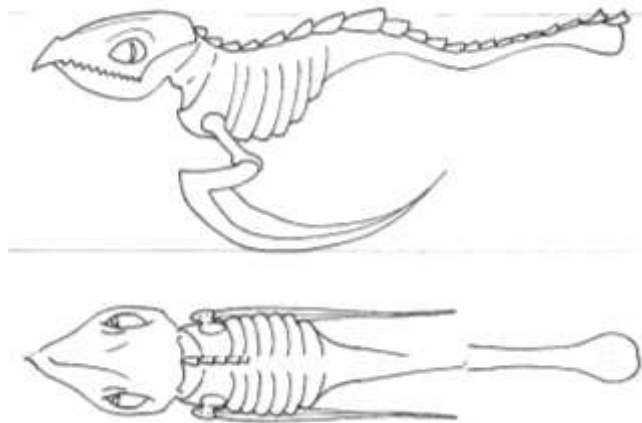


Figure 12. Offensive enemy blueprint

The final model shown in Figure 13 is composed by **1053 polys**.

## Gamification Quest: Rhythm

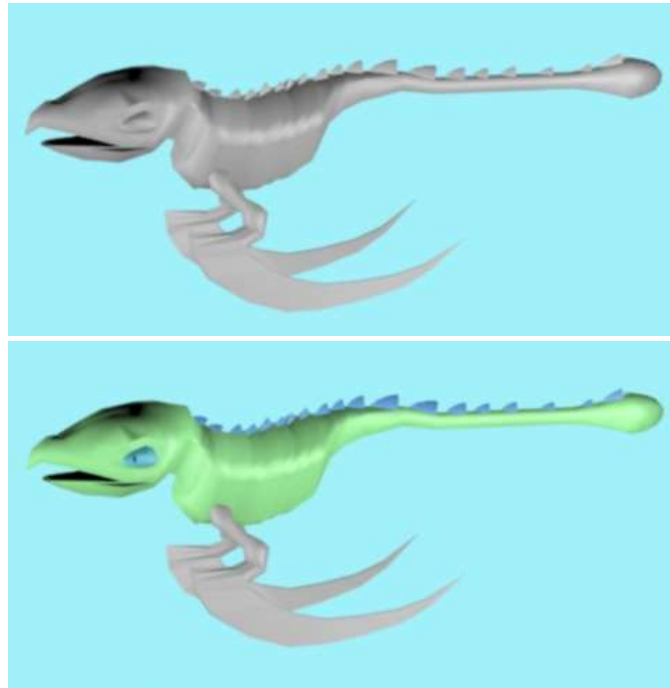


Figure 13. Final offensive enemy model with and without applied colors

### 3.4.5. A32. *Offensive enemy animation*

The previous model has been animated to keep the enemy from looking too rigid. Given that the enemy levitates over the ground, the animation recreates a sinuous movement with its body, not unlike that of eels and snakes. The model was assigned bones along its body as seen in Figure 14.



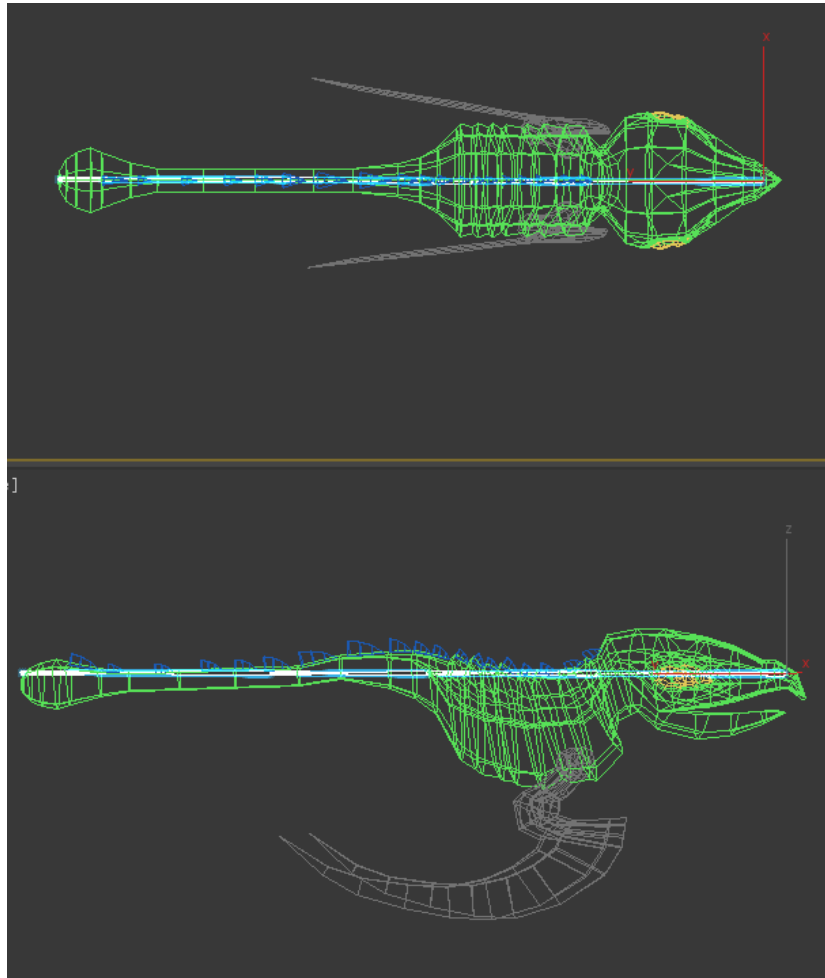


Figure 14. Bones applied to the offensive enemy 3D model

### 3.4.6. A41. Passive enemy modelling

The passive enemy model was designed but not finally included in the implementation, as described in Chapter 7. In Figure 15, the concept art shows a stocky creature, slow and with a less aggressive design than the offensive enemy design. This enemy is meant to look calm but strong, being as it will attack if the player gets too close.

Figure 16 shows the blueprint that details the shape and proportions of the most important elements in the design created.



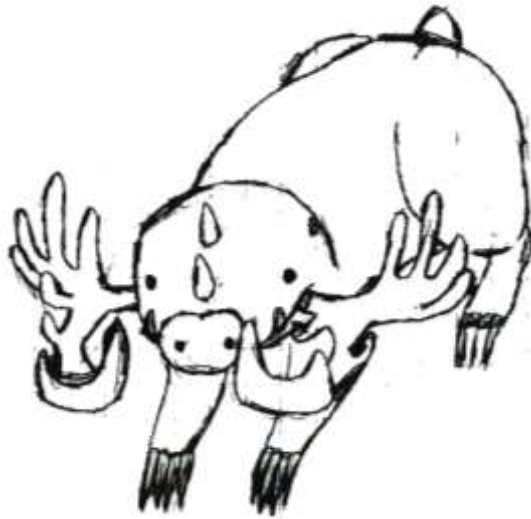


Figure 15. Passive enemy concept art

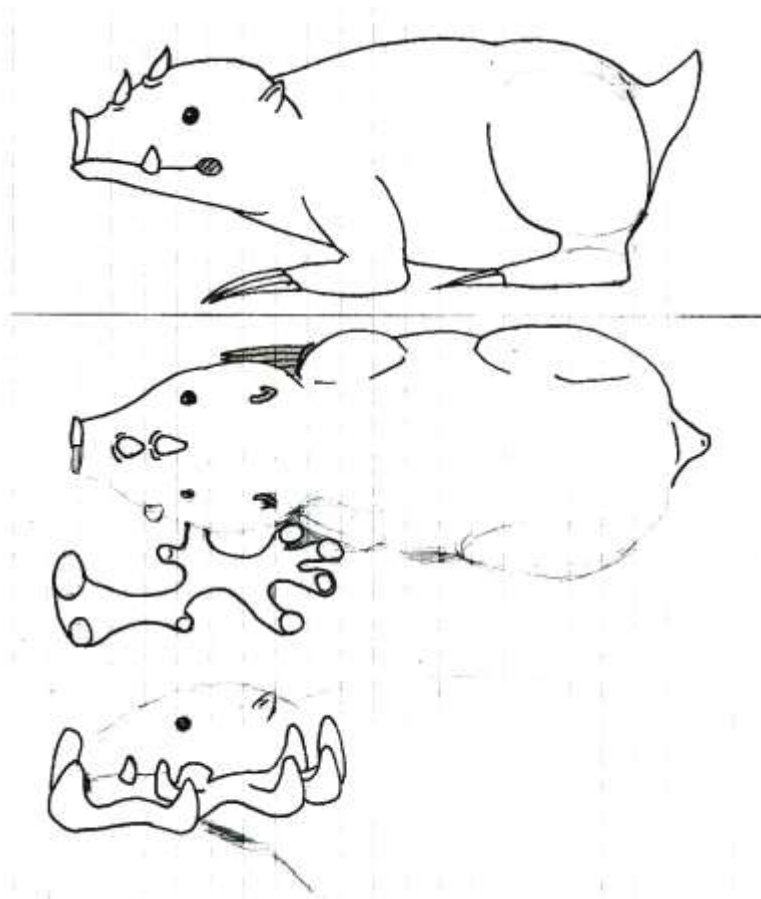


Figure 16. Passive enemy blueprint

## Gamification Quest: Rhythm


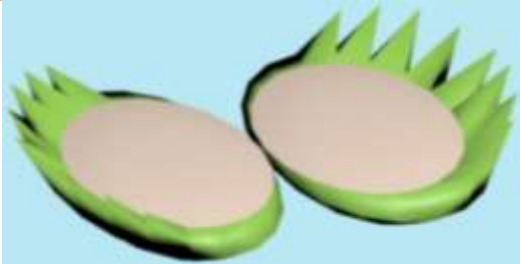

### 3.4.7. A42. Passive enemy animation


As described in Chapter 7 and Section 3.4.6, the passive enemy hasn't been modelled and thus its animation was unnecessary.

### 3.4.8. A51. Interactive objects modelling

3D models pertaining to interactive objects are included next to their polygon count in Table 11.

Table 11. Interactive objects' models

Name	Model	Polygon count
Portal		53
Carnivorous plant		410
Magnetic monolith		88

Wood plank		6
------------	--	---

### 3.4.9. A52. Interactive objects animation

Interactive objects change their behavior depending on the player's actions. While monolith's appearance changes only in color, carnivorous plants change their position from open to closed, and vice versa. This action requires a single animation with two positions, given that Unity can mirror animations. The animation was created assigning bones to the carnivorous plant's model and moving them to the needed positions in two different frames, as seen in Figure 17. 3DS Max interpolates the model's position and creates the animation as shown in the middle image. The resulting animation's speed is then adjusted in Unity.

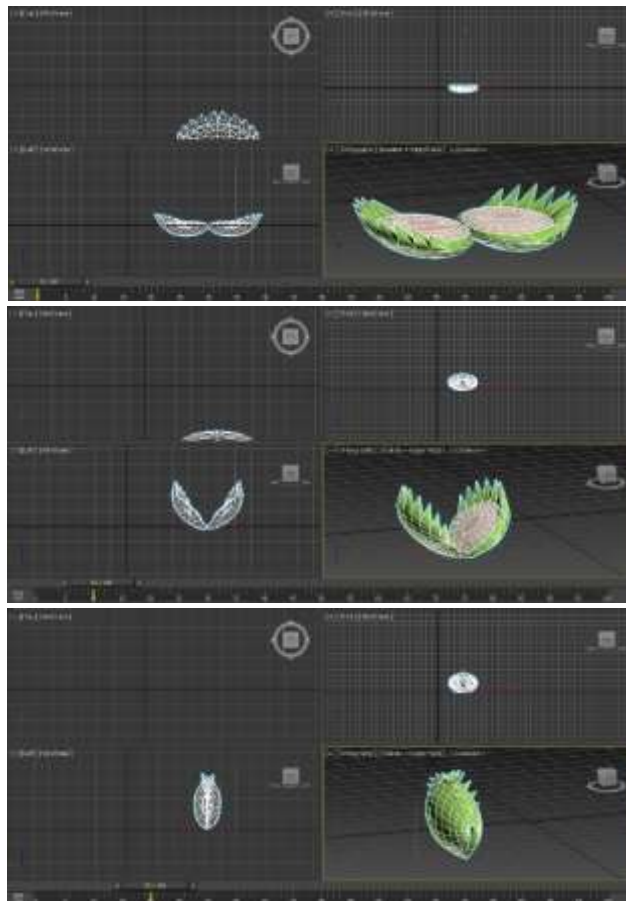


Figure 17. Three different frames of the carnivorous plant animation

## Gamification Quest: Rhythm

### 3.4.10. A6. Texturizing

The process of applying textures can be divided in two parts: creating a texture that suits the model, and correctly assigning texture coordinates to each vertex of the model. This second part is known as unwrap.

While a detailed unwrap is necessary for complex textures, it was decided early in this project that the appearance of the game would be simple, as stated in the Game Design Document. The unwrap process can be simplified this way by creating textures with coordinates that can easily be shared between multiple vertices, as is the case of a gradient. This process is detailed up next.

The textures used are composed by multiple gradients. Each model is meant to have a single texture, so when importing the model to *.fbx*<sup>3</sup> for its use in the game, a single material will be included. Gradients are beneficial because a single green gradient provides multiple green shades, so if a model includes different green tones, the texture artist can easily vary them and make modifications.

Figure 18 shows the final texture used in the familiar, which will be used to exemplify the texturizing process. This texture is only 100x100 pixels so the total model weight will be as low as possible. This is only possible thanks to the texture being composed by gradients, since color values for each vertex are interpolated and the resulting effect of applying a low-resolution version of the texture is really close to applying a high resolution one. This texture is the most complex, given that it incorporates motives and decorations in addition to the gradients. Nevertheless, the final product isn't affected by blurriness, being that the in-game model is small compared to the complete screen size and details aren't really appreciated.



Figure 18. Familiar texture

Once the texture is ready, it must be applied to a material which will be assigned to the model. Next, the process of unwrapping begins. Figure 19 shows how different parts of the model are distributed among the texture.

---

<sup>3</sup> *.fbx* is the file format used by Autodesk 3D modelling applications to export finished models and their animations. This format is supported by Unity.

# Gamification Quest: Rhythm

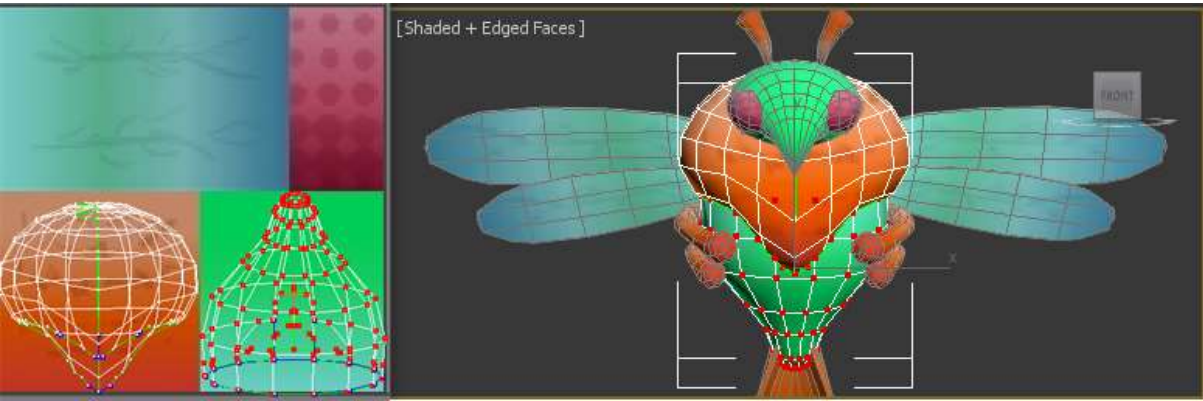


Figure 19. Distribution of vertices along the texture in the unwrap process

It is possible to recreate complex effects with only varying vertex position along the texture, as shown in Figure 20. This model emphasizes the ribs by moving the deeper creases along the gradient to a darker color. This effect can also be applied to the eyes and other details to keep the model from looking too plain.

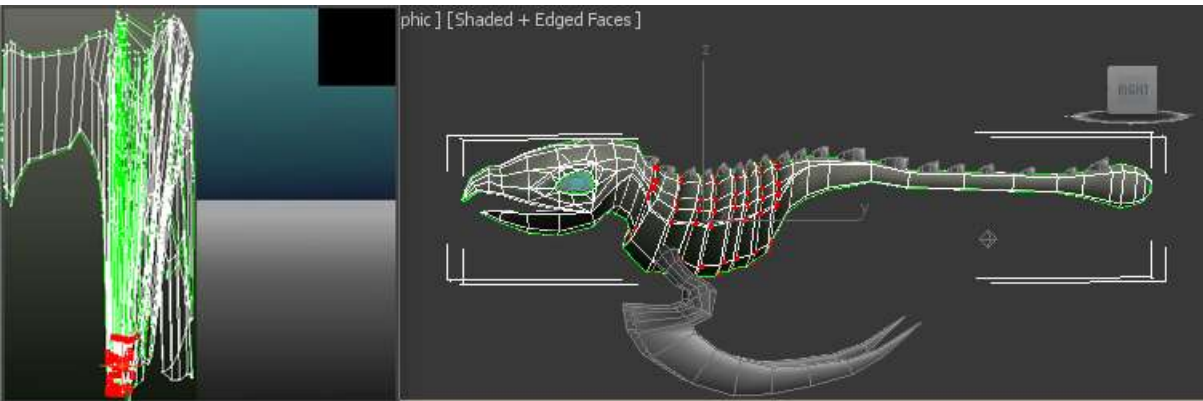




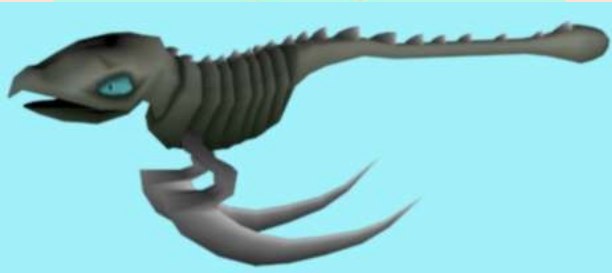


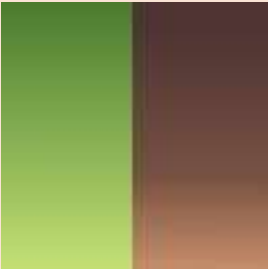



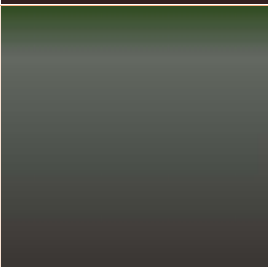
Figure 20. Creating effects with the unwrap process

In Table 12, every model is listed next to its texture. Every texture is 100x100 pixels and weights 15 Kb or less. Grass only uses part of the provided texture, given that creating a new texture with the same gradient wouldn't be efficient. Thanks to the models preserving the texture coordinates, it's possible to create variations by changing the texture, as seen on the rocks and flowers models. Some textures are shared between models, as is the case with the rocks and the portal.






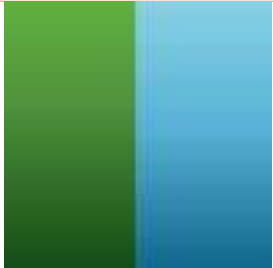

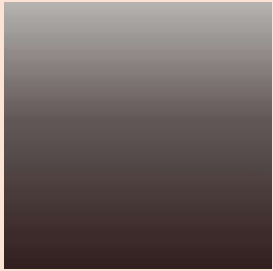
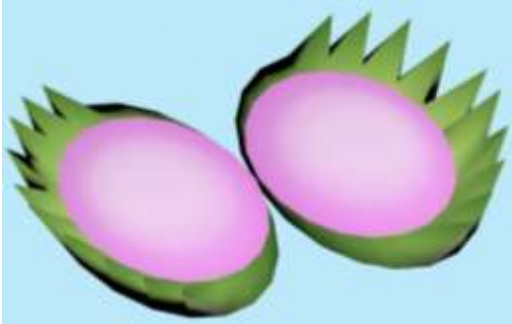
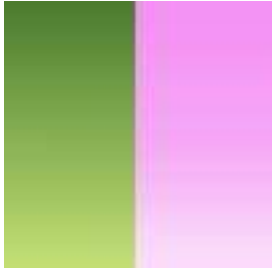
Table 12. Models with applied textures

Name	Model	Texture
------	-------	---------



## Gamification Quest: Rhythm

Familiar		
Offensive enemy		
Trees		
Rocks		
Rocks (variation)		

## Gamification Quest: Rhythm

Grass		
Flowers		
Flowers (variation)		
Portal		
Carnivorous plant		

## Gamification Quest: Rhythm

<b>Magnetic monolith</b>				
<b>Wood plank</b>				

### 3.5. General Implementation

In this section, the particular implementation in Unity to complete each task is described.

#### 3.5.1. G1. Mandolin action bar

The mandolin is played by clicking on the action zone of the instrument bar (Figure 21). The bar controls the speed and movement of the moving mark. Correct input is received when the action zone is touched once each time the mark enters the action zone. Not touching the action bar at all or touching it more than once are interpreted as fails.



## Gamification Quest: Rhythm

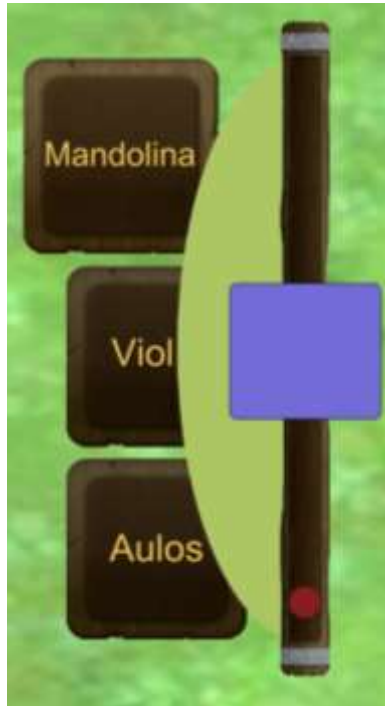


Figure 21. Mandolin instrument bar

The action zone is meant to handle touches received and act accordingly. It handles trigger detection to control whether the mark is inside or outside the action zone. A particle system is instantiated each time the player lands a successful touch in order to provide visual feedback.

### 3.5.2. G2. Violin action bar

Violin is played by keeping touch on the moving mark along the bar. This instrument bar doesn't have an action zone like the others, given that the player has to interact with the moving mark instead. This mark is much bigger in order to keep it visible even when the player keeps a finger over it, as seen in Figure 22.

In this case, there is an allowed leeway of time where the player has taken the finger off the mark until a fail is counted to keep the instrument from being too difficult to play. The bar controls if a touch is detected, and if it is, whether it's inside the mark or not.

## Gamification Quest: Rhythm



Figure 22. Violin instrument bar

### 3.5.3. G3. Aulos action bar

The aulos is an extension of the mandolin instrument bar. In this case, the player has to keep hold of the action zone as long as the mark stays inside. In Figure 23 the bigger action zone is represented. This instrument bar, as the violin instrument bar, allows a time leeway before counting a fail.



Figure 23. Aulos instrument bar

## Gamification Quest: Rhythm

### 3.5.4. G4. Instrument change

When changing instruments, the music is replaced by the version with the played instrument. There are three different music tracks, all of them including the same base music and the main melody played by the selected instrument. These tracks were composed by Victor Ávila, student from the *Conservatori Superior de Música de Castelló* (20). Information regarding the composing of these tracks can be found on Appendix B.

Tracks aren't played from the beginning each time the player selects a different instrument. All of them are the same length, and the base melody is exactly the same in all three, so replacing a track by any of the other two at the same reproduction moment results in a change in the main melody only. In order to minimize the sudden change, melodies are replaced by crossfading<sup>4</sup> the tracks.

Music keeps playing normally while the player correctly plays the selected instrument, but repeated fails or misses produce the reproduction of a fourth track. This track includes a single deep long cello note. The aim of this track is to give auditory feedback about the life they've lost by consequence of missing. When the player recovers life by correctly playing the instrument, this track fades away.

Figure 24 depicts the complete process of track reproduction at the given times.

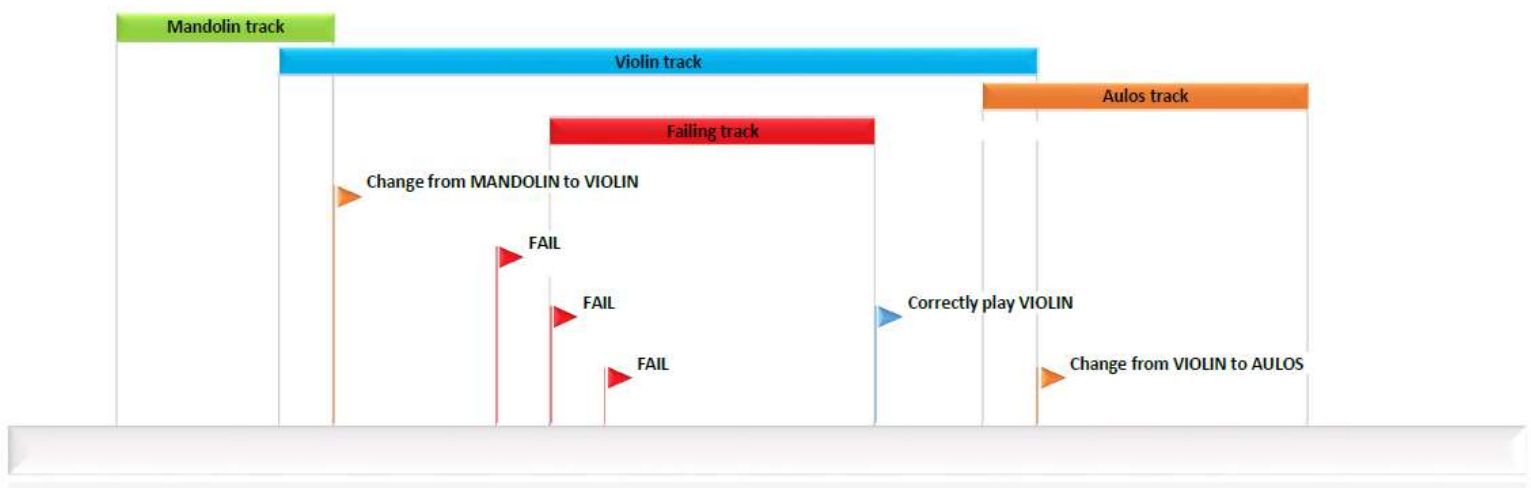


Figure 24. Example of tracks reproduced depending on the game state

Changing instruments also has other repercussions, such as modifying game elements' behaviors. The Game Controller is responsible for these modifications. Selecting an

<sup>4</sup> Crossfading refers to the process of transitioning between two audio tracks by lowering the volume of the first track at the same time the second one is raised.

## Gamification Quest: Rhythm

instrument also has graphical effects, like the selection of the new instrument's button instead of the previous one for visual feedback, and the changes described in Section 3.5.5.

### 3.5.5. G4.1. Graphical variations

The scene experiments aesthetic variations when the user selects an instrument. These variations are expressed by varying the color and intensity of the directional real time light, resulting in darker or cheerier environment. Figure 25 shows the difference between scenes caused by selecting the instruments. From top to bottom, the selected instruments are the mandolin, the violin and the aulos.



Figure 25. Graphical variations in light caused by selecting different instruments

## Gamification Quest: Rhythm

### 3.5.6. G5. Terrain creation

While the environment must give the impression of an outdoors forest, it must also limit the player's movement to guide them. For this reason, the terrain incorporates elevations and depressions that represent the borders of the level. While elevations aren't surmountable, depressions –occupied by water, as seen in Figure 26 - can be overcome by the use of abilities that allow the player to float over them.



Figure 26. Detail of terrain topology

To easily create and modify the terrain, the Unity 3D terrain editor was used. The editor enables easy modifications and it supplies tools for terrain texturizing. Figure 27 shows the generated terrain with applied textures. A grass texture was applied to the ground, while unreachable elevations were covered by sand. A dirt texture was used to indicate changes in elevation.





Figure 27. Example of terrain generated by the described method

### 3.5.7. G6. Environment setup

Once the terrain has been created, there are some configurations yet to be done before the player can correctly explore it in-game.

Environment elements were placed all along the terrain to recreate the scenery. Figure 28 shows the result of distributing these objects.



Figure 28. Terrain with environment components

## Gamification Quest: Rhythm

Each object has been previously given a collider to keep characters from going through them. In Figure 29 the collider is represented by the green box surrounding the trunk of the tree.



Figure 29. Collider applied to a tree

The first task is to define the zones in which each character can move—for example, the player can only move on the low ground, while the familiar can float and, therefore, can hover over the water. Each different zone is marked and named, and assigned to the characters that can navigate it. The creation of the navigation mesh<sup>5</sup> will also prevent characters from running into obstacles. Figure 30 shows the difference between ground level –blue– and water –red. Obstacles such as trees and rocks are excluded from the mesh.

---

<sup>5</sup> A navigation mesh, also known as navmesh, is an abstract data structure used in artificial intelligence to aid agents in pathfinding through complicated spaces (32).

## Gamification Quest: Rhythm

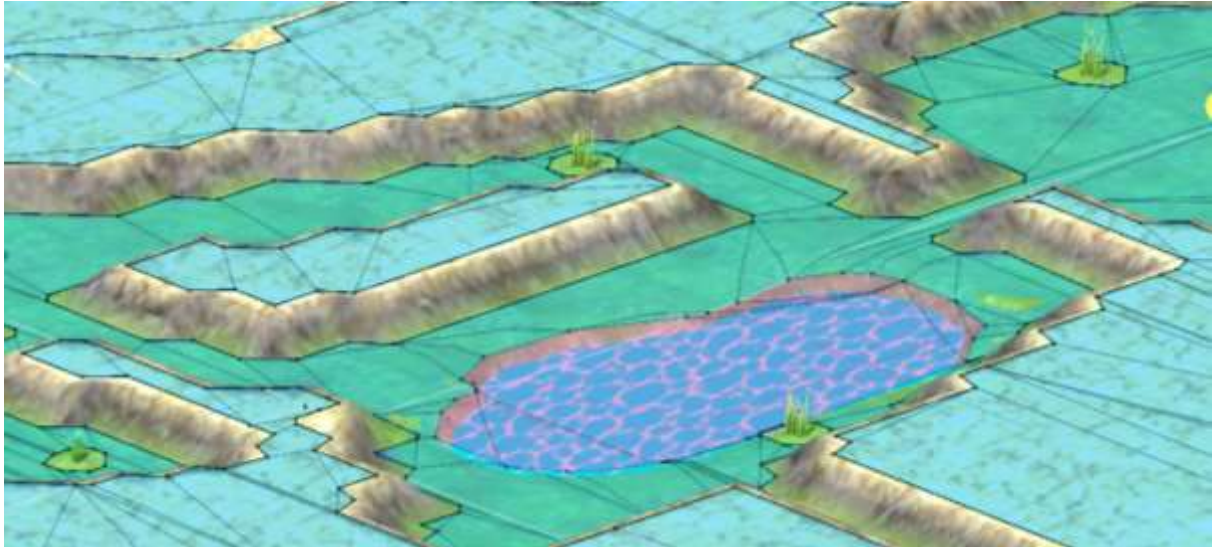


Figure 30. Generated navigation mesh for the tutorial level

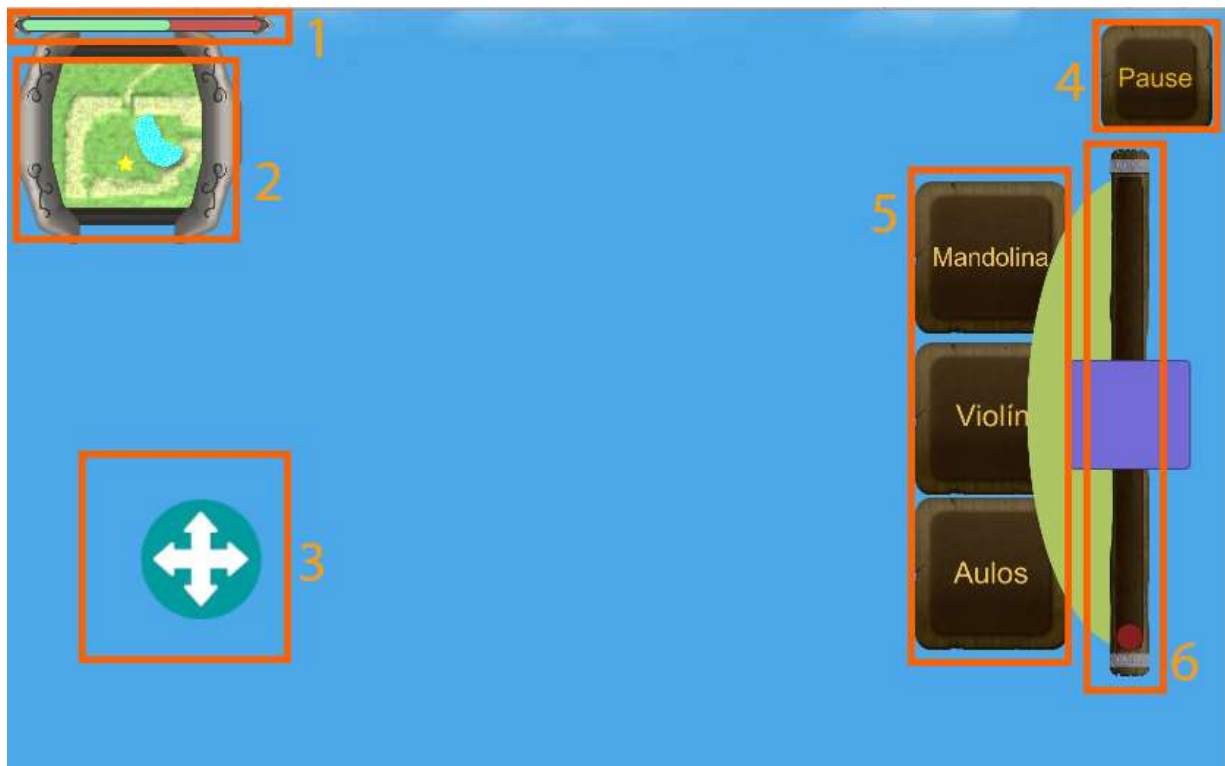
### 3.5.8. G7. User interface

The game has been implemented using a landscape orientation. The horizontal distribution fits the element's situation better in order to better control the game. The interface includes the following elements, as pictured on Figure 31.

1. **Life indicator.** Lets the player know the life level. This element is automatically updated.
2. **Mini map.** Shows a top view of the level, marking important elements such as enemies, objectives and the player's position. Holding the map enlarges it in order to better inspect it.
3. **Virtual joystick.** Controls the avatar's movement when dragged across the screen.
4. **Pause button.** Pauses the game and opens up the pause menu.
5. **Instrument buttons.** Change the selected instrument when clicked.
6. **Instrument action bar.** Allows the player to play the instrument. Varies depending on the selected instrument.

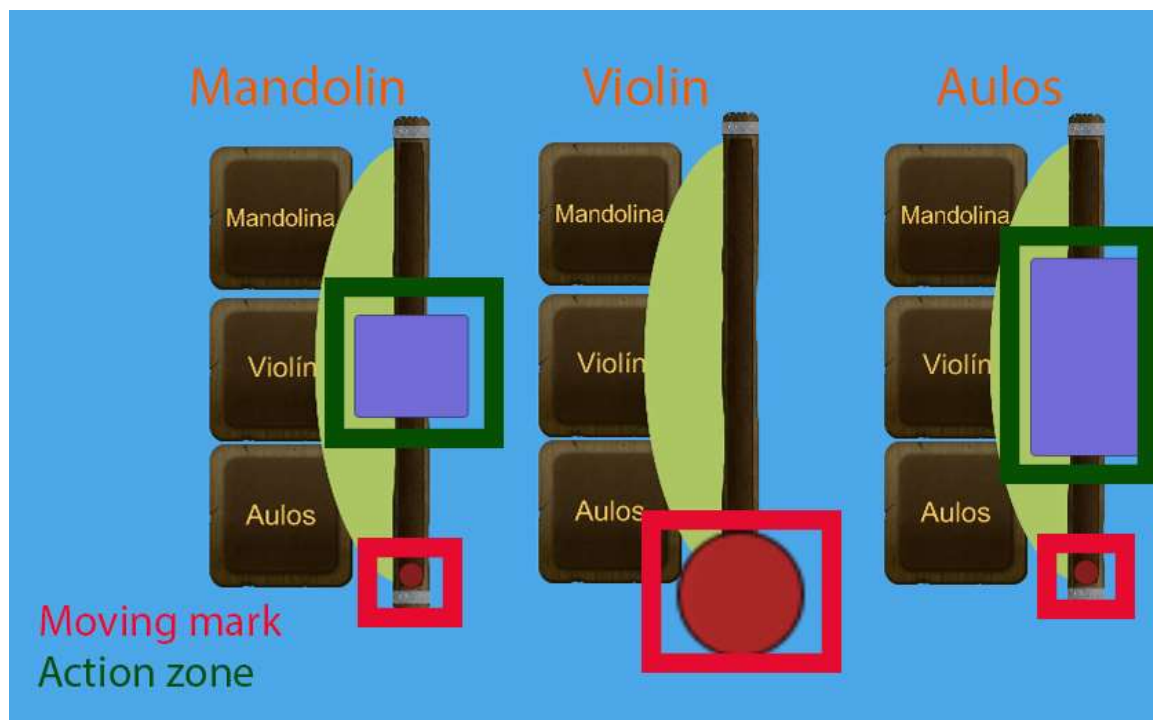


## Gamification Quest: Rhythm



**Figure 31. UI with mockup elements**

The instrument action bar is in turn composed by multiple elements depending on the selected instrument. Pictured on Figure 32, the three different action bars and their elements.



**Figure 32. Instrument action bars**

## Gamification Quest: Rhythm

The pause menu brought up by the pause button is pictured in Figure 33. This menu allows the player to change some game configurations, such as enabling or disabling tutorial mode –not implemented- and pop-up messages about the objects.



Figure 33. Pause menu

Pop-up messages give the user information about the game state or how some elements work, and appear for a limited time at given times during the gameplay. Figure 34 pictures two different scenarios in which a pop-up message might appear: on the left, approaching an interactive object; on the right, acquiring the objective.

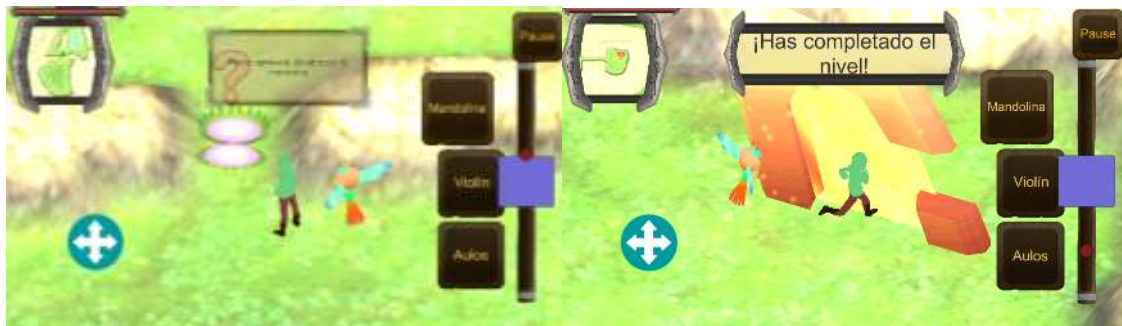


Figure 34. Pop-up messages

### 3.5.9. G8. Input setup

Given that the platform only allows tactile input, the game should be configured to work with information originating from touches on the screen. What's more, the game should be able to handle multiple touches at the same time in order to work properly.

There are multiple ways to handle touches in Unity, but version 5.6. simplified their treatment by incorporating them in the generic Event System—which handles input entries (22). In order to efficiently handle touches, the game makes use of events automatically handled by the Event System (23), *OnPointerDown* and *OnPointerClick* among them.

## Gamification Quest: Rhythm

A first approach to implement multitouch proved ineffective, being that the platform didn't, in fact, detect multiple touches at the same time. This approach used the Event System's list *Input.touches* (24), and handled each touch separately given its features. A second approach using the events first described turned out successful and touches were correctly handled by every element on screen at any time.

### 3.5.10. G9. Level graphical appearance

Materials applied to every element in the game use Unity's Standard Assets Lit Toon shader<sup>6</sup>. This shader provided by the editor produces mostly flat colors with shadows. The resulting aesthetic fits the original design described in Appendix A. Figure 35 displays one of the materials used in the game.

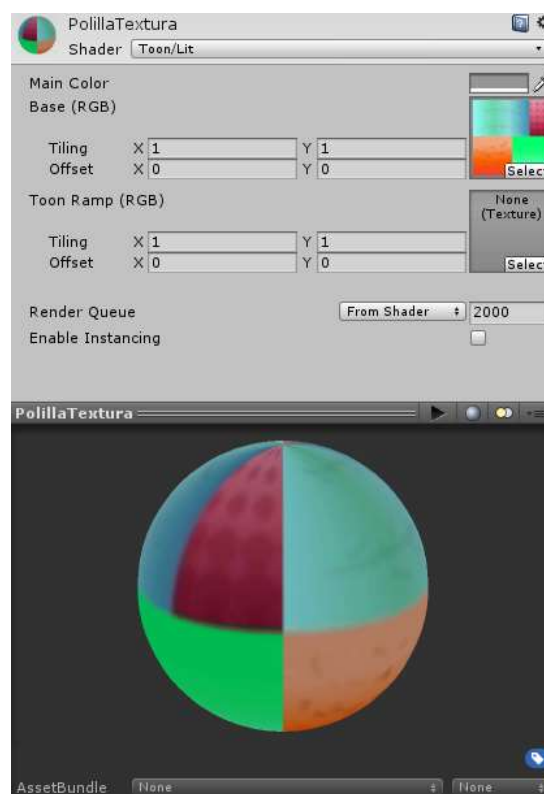


Figure 35. Example material using Toon Lit shader

The water included in the environment has been applied a material that makes use of a water shader. This shader was obtained from the Unity Asset Store, is called Water Flow FREE and was created by G.E.TeamDev (25).

---

<sup>6</sup> A shader is a computer program that produces levels of light, shadows, color and effects to be applied to an object (33).

# Gamification Quest: Rhythm

Monoliths’ force fields are represented by a sphere. This sphere uses an animated noise shader to recreate the illusion of a semi-transparent varying object. This shader is also applied to the exit portal. The shader’s code was obtained from *forestjohnson*’s contribution to the Unity Forum (26). Represented in Figure 36 are the positive (left) and negative (right) force fields.

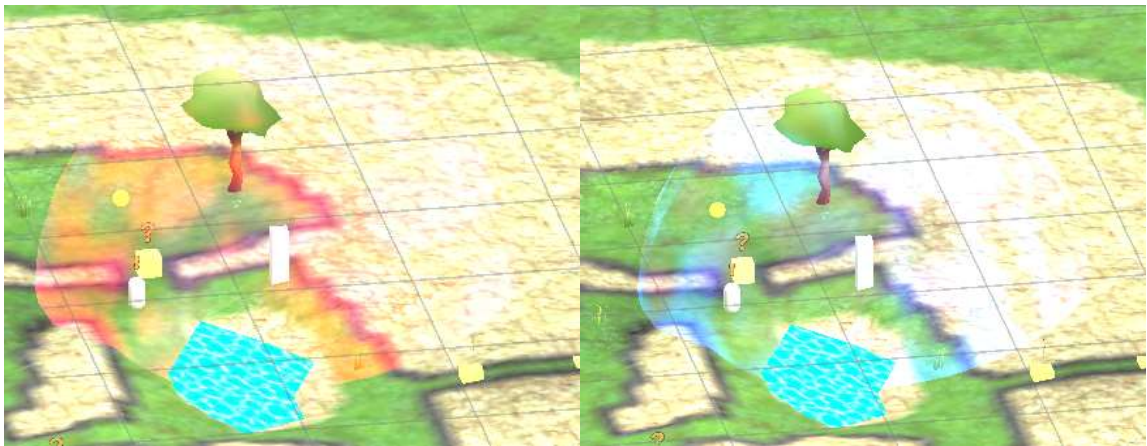



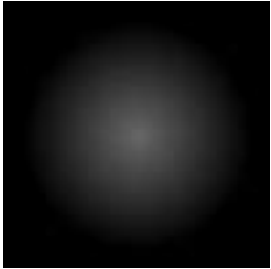

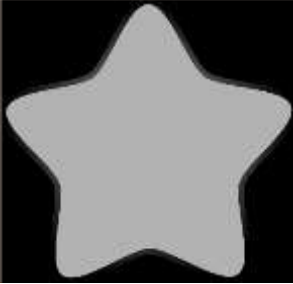



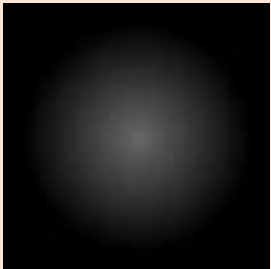
Figure 36. Magnetic force fields

Some elements of the environment include particle systems, as is the case of the exit portal and the offensive enemy. There are also UI elements that produce particles when interacting with them, like the instrument action bars. These particle systems are configured to produce different visual effects. In Table 13 renders of the produced particles are shown next to the main qualities of the particle system that produces them.

Table 13. Particle Systems found in the game

Particle System	Render	Particle	Main qualities
Offensive enemy’s tail			<ul style="list-style-type: none"><li>• Many particles</li><li>• Size variation over particle duration</li><li>• Additive particles</li><li>• Emission in a cone shape</li></ul>

## Gamification Quest: Rhythm

Exit portal			<ul style="list-style-type: none"> <li>Emission in a hemisphere shape</li> <li>Small and slow particles</li> <li>Low particle count</li> <li>Alpha decreases over particle duration</li> </ul>
Success playing instrument			<ul style="list-style-type: none"> <li>Varying rotation and size</li> <li>Low particle count</li> <li>2D emission</li> <li>Short duration</li> </ul>
Fail/Miss playing instrument			<ul style="list-style-type: none"> <li>Varying rotation and size</li> <li>Very low particle count</li> <li>2D emission</li> <li>Short duration</li> </ul>
Monolith particles			<ul style="list-style-type: none"> <li>Short duration</li> <li>Emission in a hemisphere shape</li> <li>Concentrated on the emission center</li> </ul>

The environment uses a directional light to recreate natural sunlight and the shadows it produces. This light is configured as “mixed”, which in unity means it will be used both as a real time light source and static objects light baking.



## Gamification Quest: Rhythm

To minimize real time processing of lights, most of the environment has been configured as static. A lightmap<sup>7</sup> is then baked<sup>8</sup> using the directional light. The resulting lightmap weights 6MB and stores data for all the objects on the scene. Real time light is only calculated for those objects not marked as static, such as the player or other moving elements. Light color changes depending on the instrument used, as described in Section 3.5.5.

### 3.5.11. G10. Player & camera movement

The player movement is contained in a two-dimensional horizontal plane –they can't jump or fall down through holes on the ground. The avatar's movement is controlled by the virtual joystick described in Section 3.5.8., that produces two different movement axes (X, Z) with values from 0 to 1 –where 0 means no movement and 1 means moving at the highest speed allowed. These values are treated to obtain a resulting movement vector by performing a vector addition and normalizing the resulting vector, which is then applied to the avatar in-game. This process is shown in Figure 37, where the middle image shows the default joystick position and the player stands still.

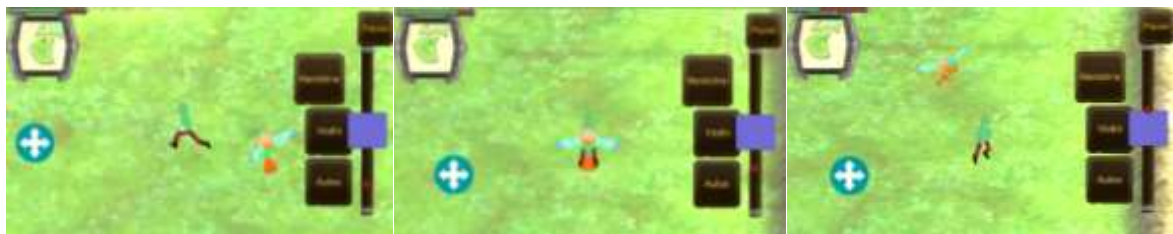


Figure 37. Three joystick positions and their effects on the avatar's movement

The avatar model is animated to produce visual feedback, and changes between an idle animation and a walking animation. When the vector obtained in the previously described method equals zero, the avatar reproduces the idle animation. Otherwise, the reproduced animation is the walking one. The speed of the walking animation is always multiplied by the magnitude of the movement vector, resulting in a slow walk when the player barely moves the joystick and a run when it's moved to its extent.

The camera is configured as orthographic, meaning it will not deform the observed scene to apply a perspective. Its rotation is fixed at a 40 grades angle to the ground and moves with the player at a fixed height and distance. Figure 38 shows the positioning of the camera during the game.

---

<sup>7</sup> A lightmap is a data structure used in *lightmapping*, a form of surface catching in which the brightness of surfaces in a virtual scene is pre-calculated and stored in texture maps for later use (34).

<sup>8</sup> Baking refers to the performance of an expensive calculation offline and then storing the obtained data in a texture map (35).

## Gamification Quest: Rhythm

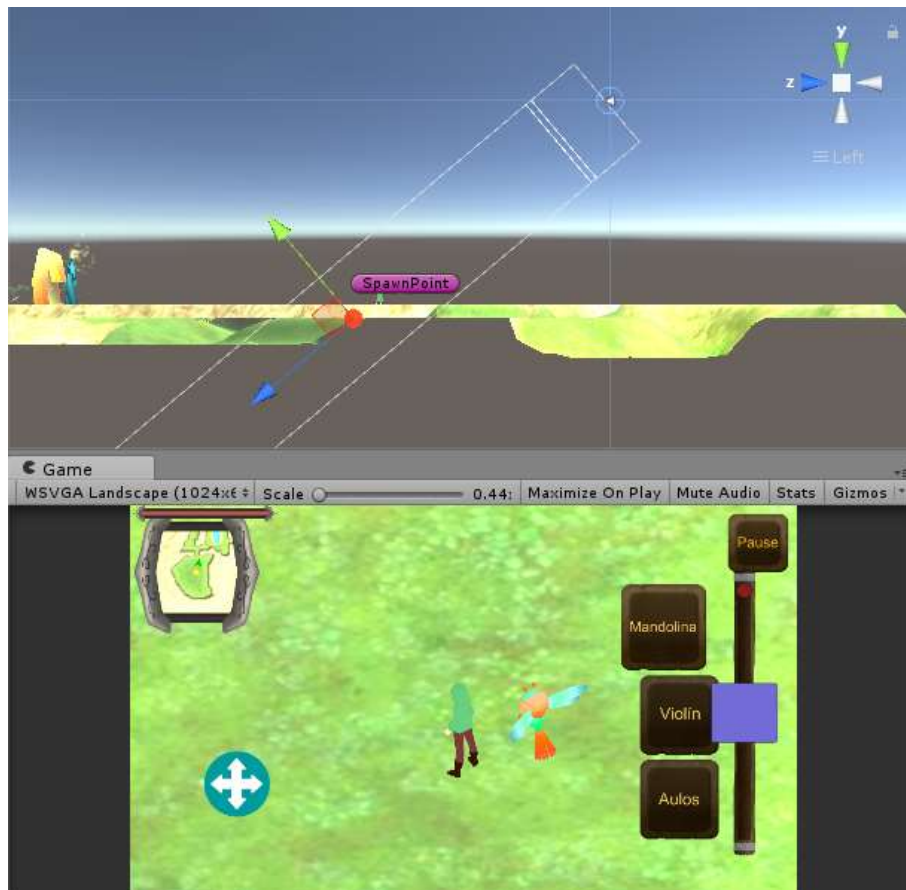


Figure 38. Camera position in-game

### 3.5.12. G11. Interactive objects response

Carnivorous plants and magnetic monoliths change their behavior depending on the selected instrument and other factors.

Carnivorous plants deactivate their colliders to let trough the player if they're playing the mandolin, and activate it otherwise. An exception is an open carnivorous plant in which the familiar has placed a wood plank. These plants won't close even when the player selects any other instrument. The plants' looks change to indicate if they're open or closed.

Magnetic monoliths, on the other side, activate a magnetic field that attracts the player to them or repulses them in the opposite direction depending on the polarity. This field is graphically represented by a semi-transparent sphere around the monolith that marks the range in which it is effective. Monoliths generate a force over the avatar once it enters its range.

The monolith also changes color depending on its polarity, which can be inversed by playing the violin. Color red indicates a positive force field, and blue a negative one. The visual representation of the force field in the form of a sphere only appears once the field has been

## Gamification Quest: Rhythm

activated by playing the aulos, as detailed in Chapter 2 and Appendix A. Figure 39 shows the looks of both polarities.

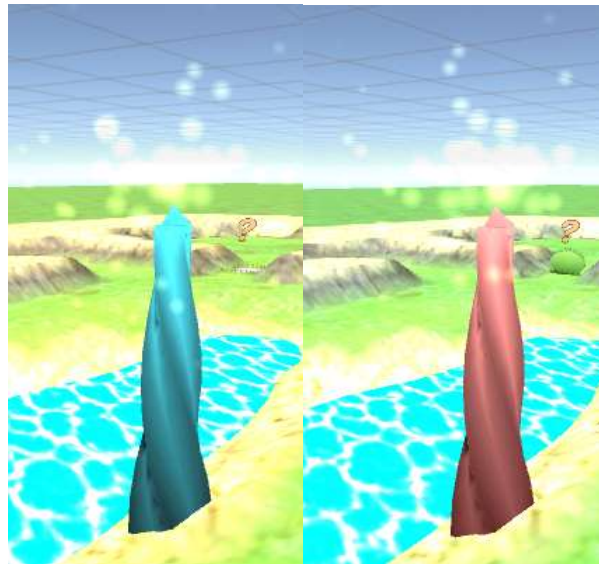


Figure 39. Negative and positive monoliths

The level exit is represented by a portal. This portal triggers the end of the level once the player has retrieved the objective and approaches it. All controls are disabled during the end sequence, in which the player automatically comes close to the portal and the scene fades out.

Wood planks can be found on certain points of a level. The familiar can pick these planks up by using the retrieve mode, and then interact with an open carnivorous plant while holding the plank in order to block the plant open. The plank will only keep the plant open for a limited period of time. Passed this period, the plant will spit the plank and close automatically if the selected instrument is not the mandolin. Figure 40 shows the described process.



Figure 40. Process of using a wood plank to keep open a carnivorous plant



## Gamification Quest: Rhythm

The game offers the player the possibility of getting information about the interactive elements. This option can be turned off on the pause menu. The provided information pops up on the top middle of the screen during a couple of seconds after the player approaches the object, and offers tips about the its use. Every interactive element has a semi-transparent question mark hovering over it while the tips are activated.

### 3.6. Artificial Intelligence Implementation

This section describes the intelligent behavior of the NPCs included in the designed game.

#### 3.6.1. AI1. Familiar AI

The familiar follows the player by default, but always follows one of the three possible behaviors depending on the selected instrument. The familiar can retrieve objects, interact with the environment and defend the player against enemies.

The familiar's movement is a combination of steering behavior and Unity's predefined navigation system. While the position of the familiar is directed by the steering behavior, the navigation agent prevents the familiar from constantly colliding against objects. The implemented steering behavior moves the familiar towards a target in a natural way, instead of moving in a straight line, to simulate an intelligent behavior. Figure 41 shows the path followed by the familiar thanks to the steering behavior.



Figure 41. Path defined by the steering behavior

## Gamification Quest: Rhythm

Table 14 includes all the possible behaviors adopted by the familiar depending on the instrument and their effects. In all cases, the familiar will always follow the player when in lack of possible actions. Figure 42 represents the decision tree that gathers all these behaviors.

Table 14. Familiar's behaviors descriptions

Behavior	Instrument played	Effects
Interact	Mandolin	If there is an interactive object within the action range and the requirements are met, interact with the object
Retrieve	Violin	If there is an object within the action range, retrieve it. If there are multiple objects, pick the closest one
Defend*	Aulos	If there are offensive enemies within the action range, attack them

\*This behavior hasn't been implemented and can't be found in the game, being that the enemy with which this behavior interacts hasn't been included in the final implementation.

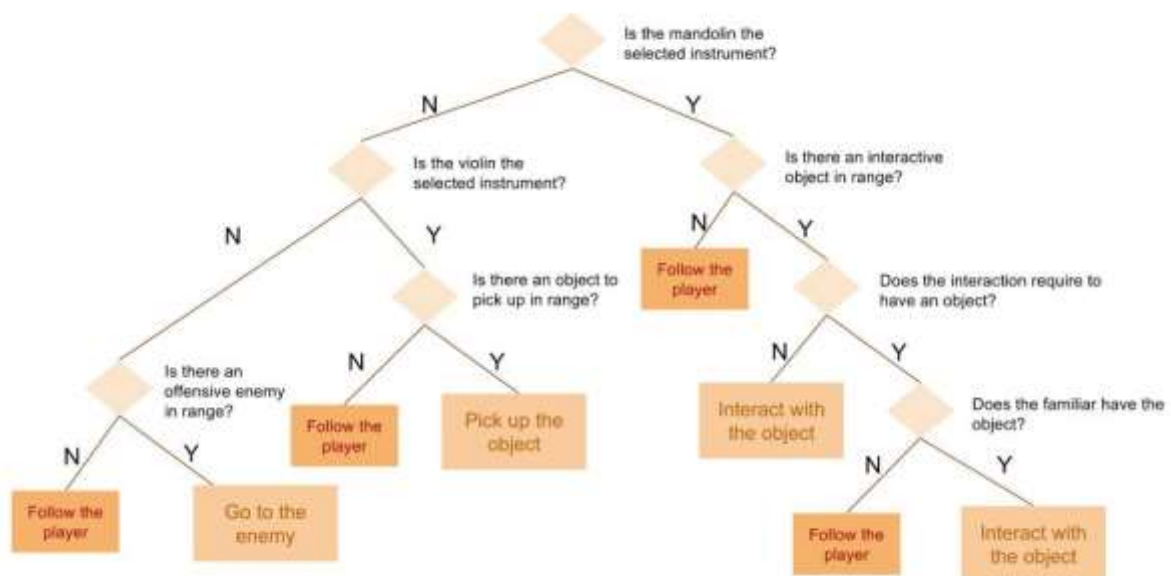


Figure 42. Familiar's decision tree

In the case of the interact behavior, some elements might require the familiar to hold another object to interact with them. This is the case of carnivorous plants. They are interactive, but the familiar won't interact with them unless it already has picked up a plank to put over them.

## Gamification Quest: Rhythm

Objectives are retrievable objects, but they are an exception as the familiar won't pick them up. When the familiar retrieves an objective, it disappears and triggers an on-screen notification, making the exit usable and marking it in the map as the next objective.

The designed defend behavior hasn't been implemented for the project. This behavior would only affect offensive enemies, as passive enemies don't actively seek the player unless attracted by the mandolin. If the defend behavior was active, offensive enemies would seek the familiar instead of the player. This way, the familiar would receive the damage. Correctly playing the aulos would produce a successful attack on the behalf of the familiar, while failing would produce a successful enemy attack.

### 3.6.2. AI2. Offensive enemy AI

The behavior of the offensive enemy wasn't implemented in the game for this project, as stated in Chapter 7. Nevertheless, the behavior was previously designed. The decision tree that would dictate this enemy's actions is shown in Figure 43. This enemy would patrol inside a delimited zone while the player was outside its action range, and seek the player otherwise.

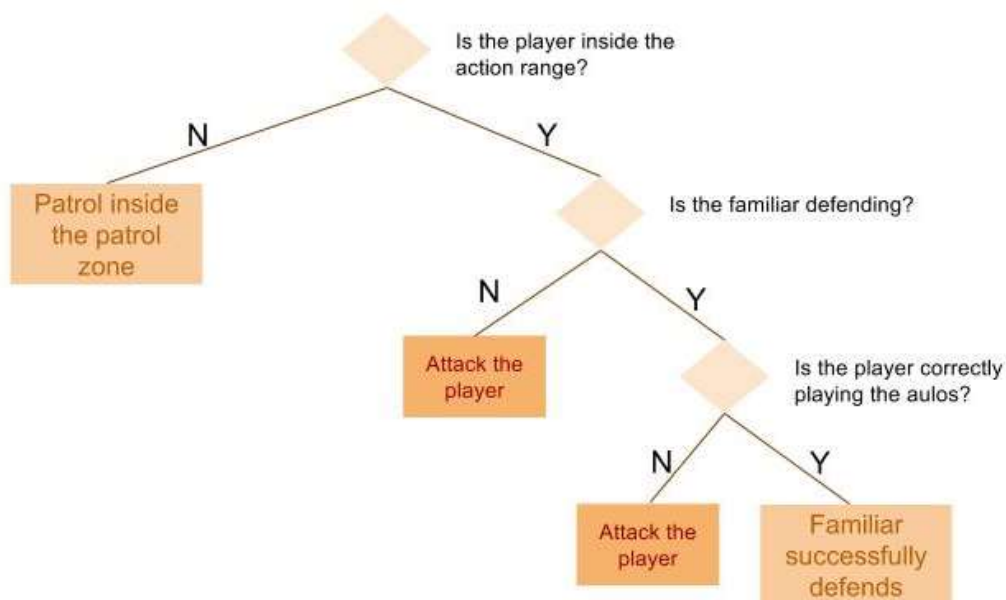


Figure 43. Offensive enemy's decision tree

### 3.6.3. AI3. Passive enemy AI

The passive enemy stays still unless the player selects the mandolin and enters its range without obstacles in between. Its behavior then changes, and seeks the player if there is a straight line between their positions. If the enemy gets too close to the player, it will hurt them. The decision tree shown in Figure 44 represents the movement behavior described;

## Gamification Quest: Rhythm

the attack logic is applied separately by checking the distance between the enemy and the player.

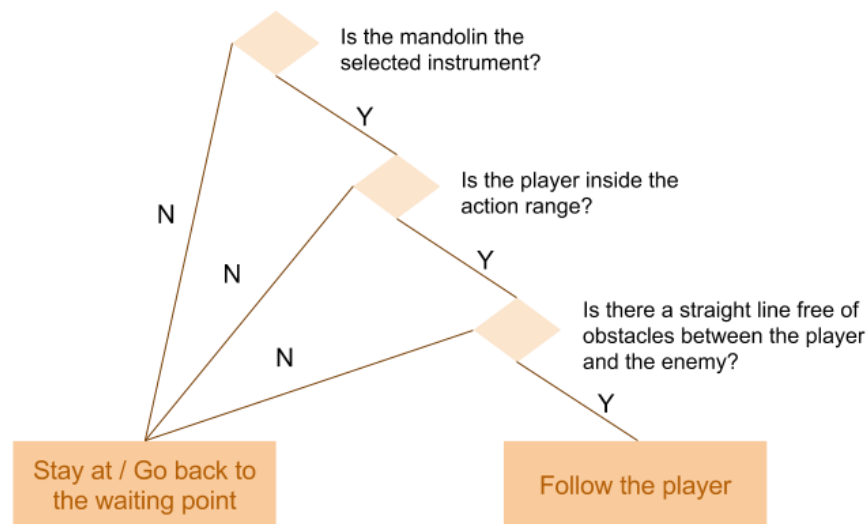


Figure 44. Passive enemy's movement decision tree

The passive enemy checks if the user is inside its range thanks to a trigger collider, and uses a Raycast<sup>9</sup> to check if there are obstacles between its position and the player's. The Raycast is created using both the enemy's position and the direction of the vector resulting of subtracting the player's position and the enemy's. A RaycastHit<sup>10</sup> is then obtained, and the enemy checks whether the object hit was tagged as the player –meaning there is a straight line free of obstacles between them- or not.

### 3.7. Testing

Feature testing was applied after every major implementation, such as the inclusion of a new AI that implied interaction between multiple game elements.

Detailed testing was performed once the project was declared complete to prevent bugs from being included in the final build of the game.

Testing is discussed on Chapter 5 in detail.

<sup>9</sup> A Raycast is, in Unity, a ray originating from a point in space and with a direction. Information can be obtained from this object regarding collisions and physics (36).

<sup>10</sup> A RaycastHit is a structure used to get information back from a Raycast (37).

# Chapter 4

## Results

---

This chapter summarizes the project's achievements and compares the results to the objectives declared in Chapter 1. The final state of the implemented game is presented.

### 4.1. Objective completion

This project's objectives were stated in Chapter 1. This section describes in which ways the final game state satisfies them.

1. *Implement game mechanics that reward students' actions, taking advantage of the gamification application in which the game is integrated.*

The designed game uses information from the gamification app such as the player's average time taken to deliver tasks and grades, and balances game elements (enemies' strength, max life points, life recovery) depending on them. The game has been successfully incorporated in the app, and both available levels (the tutorial and the first level) can be unlocked and played through it. While the tutorial is automatically unlocked when a new account is created, the first level must be unlocked.

2. *Integrate music as a game mechanic on the environment.*

The core game mechanic –playing instruments to magically control the game- works by correctly playing each instrument, which has its own melody and different playing instructions. The effects the music produces on the environment tie together the mechanics with the scenery, connecting the player's actions to the game and its behavior.

3. *Design replayable levels without modifying their structure.*

Multiple player spawn points and exit points have been introduced in the designed level to increase replayability without modifying the level's topology. The player will access different

## Gamification Quest: Rhythm

spaces on the level as needed in search of the objective, also selected by the game, to create different paths and itineraries inside a big level. Section 3.3.2 describes the elements' distribution on the map.

### 4.2. Final game state

The game obtained in the development of this project is the result of programming, modelling and design efforts. All the disciplines come together to form the final product.

The efforts made in the development resulted in:

- about 2.000 code lines implemented
- 13 3D models
- 2 animations
- a tutorial level
- a complete level

and

- the complete game design of Gamification Quest: Rhythm

The resulting game can be downloaded and installed in any Android device with a running operative system above Android 4.1 Jelly Bean (API version 16). Figure 45 redirects the device used to scan the QR code to the required .apk download. Alternatively, the archive can be accessed using the following link:

<https://drive.google.com/open?id=0BzF5zjEy35UUTUx4R0pzYXI3VTA>



Figure 45. Gamification Quest: Rhythm .apk QR code

To login, use **USER: alumno3** **PASSWORD: Alumno-3** Once logged in, select the mission menu (sword icon) and start “Tutorial” for the tutorial level and “Misión 1” for the complete level.

## Gamification Quest: Rhythm

The game implements a complete level progression (starting the level, searching the environment for the objective, acquiring it, and completing the mission by finding the exit). This progression is also included into the gamification app, meaning the player will access the game by unlocking missions. The app will manage the rewards the player can get by completing the level, and the experience they will get after the process. A complete gameplay of both implemented levels and some highlights can be found in the following link:

<https://drive.google.com/file/d/0BzF5zjEy35UUNUlxTdlIdURYdIU/view?usp=sharing>

The video was recorded on a Nexus 5 device running Android 6.0.1. Marshmallow thanks to the app described in Section 3.2.5. The game has been integrated in the Gamification Quest app, as shown in the first sequence of the video, where the player selects the playable level.

# Chapter 5

## Testing & Evaluation

---

The testing of the implemented features described in this document took place during the development phase of the project. During this period, some issues were found and resolved. The bugs with the biggest impact in the development are listed in **¡Error! No se encuentra el origen de la referencia.** next to the solutions found.

Table 15. Big impact bugs found during development

ID	Description	Solution
B01	Multi-touch not working on all the game elements	Restructuring touch detection
B02	Collisions between NPCs result in applied forces on both of them that modify their movement inconsistently	Resetting forces to zero after each collision
B03	Static objects' meshes were combined by consequence of a faulty build. This is a known issue in the Unity community (27). This resulted in faulty shadows, unmovable objects and an out-of-date <i>navmesh</i>	Manually replace all combined meshes by their original mesh

Further testing is necessary once the game has been completely implemented to prevent minor bugs from making it into the final build. No other major bugs were found. Nevertheless, playtesting sessions are necessary not only to find bugs, but to modify design decisions and adjust the game's difficulty. The questionnaire included in Appendix C includes the questions relevant to gather testers' opinions and experiences, and make changes taking the results into account. The questions included regard multiple game components, such as playability, graphics, sound design and the overall experience.



# Chapter 6

## Conclusions

---

This chapter summarizes the content of this document, the learnt competences developed thanks to the project, and possible future work for the game created.

### 6.1. Project overview

This document describes the project *Gamification Quest: Rhythm, music as a game mechanic* and details the development and the obtained results.

The Technical Proposal, included in Chapter 1, declares the objectives for this project, describes its contents and contemplates the initial planning and task duration for the needed work. This chapter also includes information about the motivation behind the project, the skills required for its completion and the expected results.

Following the Technical Proposal, a summary of the Game Design Document included in Appendix A was provided in Chapter 2. This design contemplates functionalities not implemented for this project, being that the document details every element included in a complete version of the game designed.

The description of the process followed to complete each task described in the Technical Proposal is then detailed in Chapter 3. This chapter extensively describes the implementation and working of every game element implemented for the project. Chapter 4 describes the fulfillment of the objectives stated in the Technical Proposal, and describes the game created, providing context for the features implemented. An overview on the testing of these features is then discussed in Chapter 5.

From this description and the results described in Chapter 4, we can confirm the successful completion of this project.

## Gamification Quest: Rhythm

### 6.2. Developed competences

The completion of this project required the use of skills acquired both during the bachelor's degree and learnt separately. Working on this project on a tight schedule required individual organizational skills not practiced to the extent demanded until this point in the degree. The wrong estimation of task duration was caused by the lack of experience in the planning of long term projects of this size and demand level. Thanks to the project and the experience acquired, future work will most surely be better organized.

This project also helped in developing better fluidity when working with Unity. The constant use of this game engine in the development industry indicates that experience in the use of the engine will continue being in high demand in the following years (28) (29).

The opportunity to completely design a game and its mechanics, and the implementation of these mechanics, followed by the consequent testing, provides needed experience in the field of game development. Sometimes, the difference between well designed mechanics and failure only resides in experience.

### 6.3. Future work

The designed game could see some improvement in the extension of interactive elements. Currently, the game is prepared to implement objects with different effects in the game, besides carnivorous plants and magnetic monoliths, wood planks and enemies. Some of the functionalities described in Appendix A and Chapter 1 haven't been implemented, as stated in Chapter 7. The gamification app in which the game is integrated would also benefit from the implementation of multiple levels.

The project could be extended as a whole with the upgrade of the instruments' action bars. The upgrade would consist in the automatic adaption of the rhythm in which it moves to the melody produced by the instrument, contrary to the manual adjustment of a constant rhythm as currently implemented.

Further advancements could also be made in the ways of gamification. Designing new ways in which to connect the game and the advance in education of the player could benefit the player's motivation and performance in the educational environment.

# Chapter 7

## Project Deviations

During the development, the original project plan was modified to fit the project's requirements. As stated in Section 1.7, a risk that the distributed time for each task wasn't correctly estimated existed (R1). Objectives' completion was a priority, and so some tasks were left behind when essential tasks proved more demanding than contemplated. Table 16 states the final time consumption for each task.

Table 16. Project plan variations

ID	Task name	Initial time estimate	Final time estimate
D1	Technical Proposal	10	25
D2	Game Design Document	18	25
D3	Technical Report	25	65
D4	Presentation Video	1	2
D5	Final Presentation	5	7
GD1	Game mechanics	15	20
GD2	Level design	3	8
A1	Environment	20	8
A21	Familiar modelling	5	8
A22	Familiar animation	5	-
A31	Offensive enemy modelling	5	5
A32	Offensive enemy animation	5	0.5*
A41	Passive enemy modelling	5	-
A42	Passive enemy animation	5	-
A51	Interactive objects modelling	5	4
A52	Interactive objects animation	5	0.5
A6	Texturizing	5	5
G1	Mandolin action bar	7	4
G2	Violin action bar	7	4
G3	Aulos action bar	7	4

## Gamification Quest: Rhythm

<b>G4</b>	Instrument change	5	8
<b>G41</b>	Graphical variations	15	1*
<b>G5</b>	Terrain creation	5	10
<b>G6</b>	Environment setup	8	7
<b>G7</b>	User interface	4	3
<b>G8</b>	Input setup	2	7
<b>G9</b>	Level graphical appearance	15	8
<b>G10</b>	Player & Camera movement	5	3
<b>G11</b>	Interactive objects response	15	10
<b>A11</b>	Familiar AI	25	20
<b>A12</b>	Offensive enemy AI	10	-
<b>A13</b>	Passive enemy AI	10	3
<b>T1</b>	Feature testing	5	7
<b>T2</b>	Complete game testing	3	3
<b>T3</b>	Post-Testing modifications	10	15
		<b>TOTAL</b>	300

5: Less time required to complete task

5: Time required was correctly estimated

5: More time required to complete task

5\*: Task completed with restrictions

-: Task not completed

Some tasks took more time than estimated to complete. The most notorious case is that of the technical report, which took more than two times the time estimated.

Other tasks were not completed to the extent of the initial design described in Appendix A. These tasks (A32, G41) and the tasks discarded (A22, A41, A42, A12) were not considered essential for the completion of the objectives stated.

Animations were reduced to the essential ones, making up for them with scripts in Unity where possible. This is the case of the familiar model, whose animations (A22) were replaced by a script that made the model float up and down. The offensive enemy's animations (A32) were reduced to an idle movement loop. Being that the passive enemy didn't levitate like the previously mentioned models, dropping the modelling altogether (A41) was the best option, given that to achieve an acceptable graphical appearance the model needed at least two different animations (idle, and walking) to differentiate between its movement states (A42).

The original game design described in Appendix A described graphical variations (G41) affecting not only the light, but also the 3D models and materials. These graphical changes were considered irrelevant to the completion of the objectives and were finally reduced to light variations.

## Gamification Quest: Rhythm

The passive enemy's AI(AI2) was considered the most relevant from a game mechanics perspective. While implementing the offensive enemy's AI(AI2) would be interesting for the level design, it wasn't necessary, as the existence of passive enemies provided sufficient confrontation. This task was finally dropped to prioritize essential tasks such as the level and mechanics implementation.

Thanks to some tasks taking less time than estimated, the project has been completed satisfactorily.

# Chapter 8

## References

---

1. **Julián Moreno Cadavid, Luis F. Montoya Gómez.** Uso de un entorno virtual de aprendizaje ludificado como estrategia didáctica en un curso de pre-cálculo: Estudio de caso en la Universidad Nacional de Colombia. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*. [Online] 2015. [Cited: Jan 28, 2017.] [http://www.scielo.mec.pt/scielo.php?script=sci\\_arttext&pid=S1646-98952015000400002](http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S1646-98952015000400002).
2. **Díaz Cruzado, Jesús y Troyano Rodríguez, Yolanda.** El potencial de la gamificación aplicado al ámbito educativo. [Online] 2015. [Cited: Jan 28, 2017.] [https://fcce.us.es/sites/default/files/docencia/EL%20POTENCIAL%20DE%20LA%20GAMIFICACI%C3%93N%20APLICADO%20AL%20C3%81MBITO%20EDUCATIVO\\_0.pdf](https://fcce.us.es/sites/default/files/docencia/EL%20POTENCIAL%20DE%20LA%20GAMIFICACI%C3%93N%20APLICADO%20AL%20C3%81MBITO%20EDUCATIVO_0.pdf).
3. **Stepmania.** [Online] [Cited: Jun 19, 2017.] <https://www.stepmania.com/>.
4. **Ubisoft.** Rayman Legends. [Online] [Cited: Jun 19, 2017.] <https://www.ubisoft.com/es-es/game/rayman-legends/>.
5. **Crypt of the Necrodancer.** [Online] [Cited: Jun 19, 2017.] <http://www.necrodancer.com/>.
6. **Office Timeline.** [Online] [Cited: Jun 19, 2017.] <https://www.officetimeline.com/>.
7. **Game Career Guide.** *Game Developer Salary Survey 2013*. [Online] 2013. [Cited: Jun 21, 2017.] [http://www.gamecareerguide.com/features/1279/game\\_developer\\_salary\\_survey\\_.php?page=2](http://www.gamecareerguide.com/features/1279/game_developer_salary_survey_.php?page=2).
8. **KDE.** Krita. [Online] [Cited: Jun 19, 2017.] <https://krita.org/en/>.
9. **Audacity Audio Editor.** [Online] [Cited: Jun 19, 2017.] <http://audacity.es/>.

## Gamification Quest: Rhythm

10. Unity. [Online] [Cited: Jun 19, 2017.] <https://unity3d.com/es/>.
11. JavaScript Programming Language. [Online] [Cited: Jun 19, 2017.] <https://www.javascript.com/>.
12. C# Programming Language. [Online] [Cited: Jun 19, 2017.] <https://docs.microsoft.com/es-es/dotnet/csharp/csharp>.
13. Boo Programming Language. [Online] [Cited: Jun 19, 2017.] <http://boo-lang.org/>.
14. Adobe. Photoshop. [Online] [Cited: Jun 19, 2017.] <http://www.adobe.com/es/products/photoshop.html>.
15. Autodesk. 3DS Max. [Online] [Cited: Jun 19, 2017.] <https://www.autodesk.com/products/3ds-max/overview>.
16. Microsoft Office. Word. [Online] [Cited: Jun 19, 2017.] <https://products.office.com/es-es/word>.
17. Writelatex Limited. Overleaf. [Online] [Cited: Jun 19, 2017.] <https://www.overleaf.com/>.
18. Microsoft Office. PowerPoint. [Online] [Cited: Jun 19, 2017.] <https://products.office.com/es-es/powerpoint>.
19. TechSmith. Camtasia. [Online] [Cited: Jun 19, 2017.] <https://www.techsmith.com/video-editor.html>.
20. Conservatori Superior de Música de Castelló (CSMC). [Online] [Cited: Jun 20, 2017.] <http://conservatorisuperiorcastello.com/index.html>.
21. Hecorat. AZ Screen Recorder. [Online] <https://play.google.com/store/apps/details?id=com.hecorat.screenrecorder.free>.
22. Unity Manual. [Online] [Cited: Jun 17, 2017.] <https://docs.unity3d.com/Manual/script-TouchInputModule.html>.
23. Unity Manual. [Online] [Cited: Jun 17, 2017.] <https://docs.unity3d.com/Manual/SupportedEvents.html>.
24. Unity Documentation. [Online] [Cited: Jun 20, 2017.] <https://docs.unity3d.com/ScriptReference/Input-touches.html>.
25. Unity Asset Store. [Online] [Cited: Jun 22, 2017.] <https://www.assetstore.unity3d.com/en/#!/content/26434>.

## Gamification Quest: Rhythm

26. Unity Forum. [Online] [Cited: Jun 22, 2017.] <https://forum.unity3d.com/threads/fx-force-field-shader.9535/>.
27. Unity Issue Tracker. [Online] [Cited: Jun 27, 2017.] <https://issuetracker.unity3d.com/issues/gameobjects-mesh-is-stuck-at-combined-mesh-root-scene-and-cannot-be-moved-after-aborting-a-build-process>.
28. G2 Crowd Game Engine Statistics. [Online] [Cited: Jun 28, 2017.] <https://www.g2crowd.com/categories/game-engine>.
29. The Next Web. *This engine is dominating the gaming industry right now*. [Online] [Cited: Jun 28, 2017.] <https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/>.
30. Wikipedia. [Online] [Cited: Jun 05, 2017.] <https://en.wikipedia.org/wiki/Roguelike>.
31. Wikipedia. [Online] [Cited: Jun 05, 2017.] [https://en.wikipedia.org/wiki/Familiar\\_spirit](https://en.wikipedia.org/wiki/Familiar_spirit).
32. Wikipedia. [Online] [Cited: Jun 20, 2017.] [https://en.wikipedia.org/wiki/Navigation\\_mesh](https://en.wikipedia.org/wiki/Navigation_mesh).
33. Wikipedia. [Online] [Cited: Jun 20, 2017.] <https://en.wikipedia.org/wiki/Shader>.
34. Wikipedia. [Online] [Cited: Jun 22, 2017.] <https://en.wikipedia.org/wiki/Lightmap>.
35. Wikipedia. [Online] [Cited: Jun 22, 2017.] [https://en.wikipedia.org/wiki/Glossary\\_of\\_computer\\_graphics#baking](https://en.wikipedia.org/wiki/Glossary_of_computer_graphics#baking).
36. Unity Documentation. [Online] [Cited: Jun 26, 2017.] <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>.
37. Unity Documentation. [Online] [Cited: Jun 26, 2017.] <https://docs.unity3d.com/ScriptReference/RaycastHit.html>.

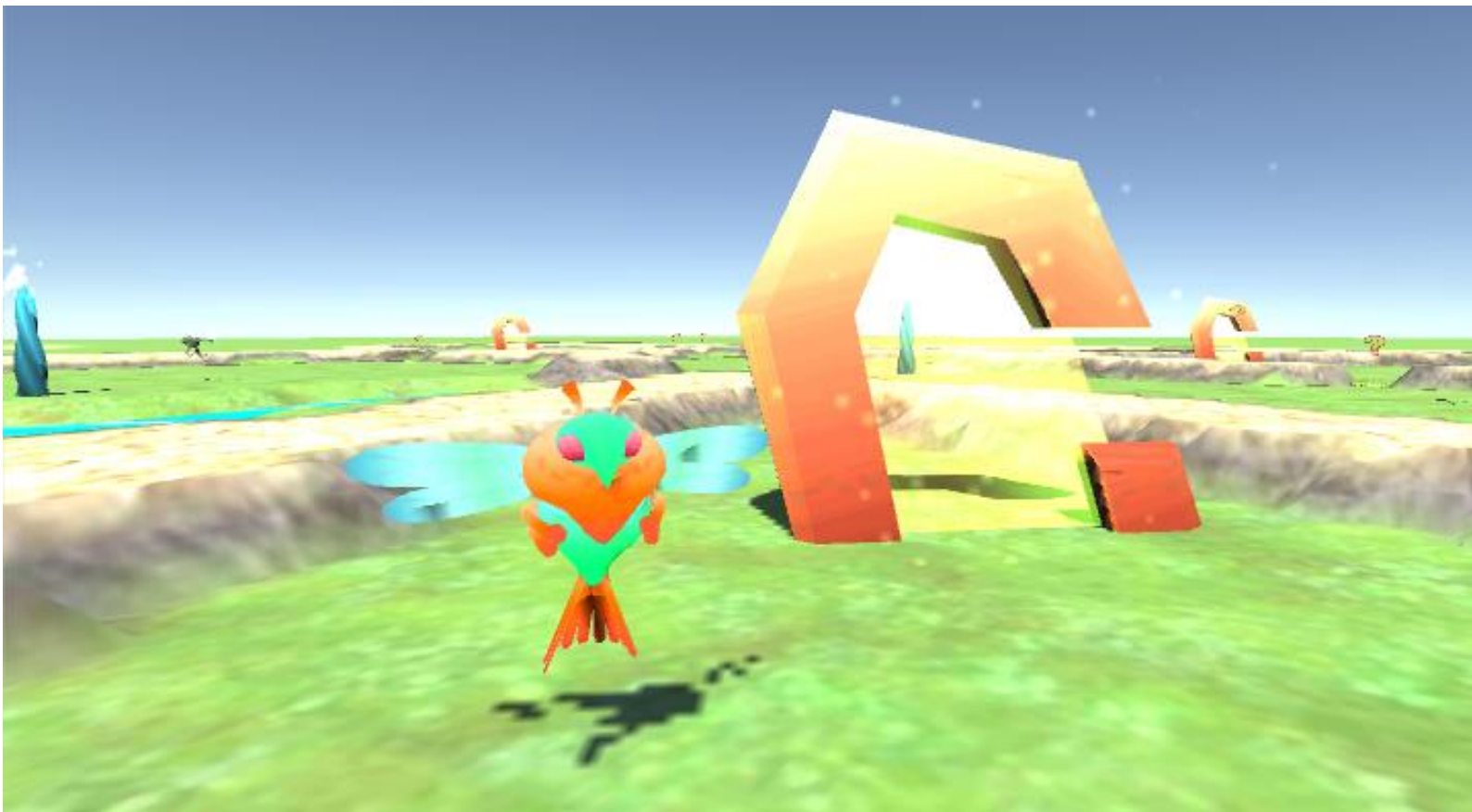


# Chapter 9

## Appendix A.

Gamification Quest: Rhythm GDD

---



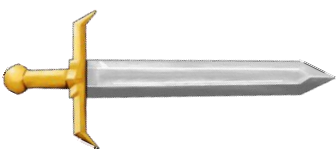
- Game Design Document -

# Gamification Quest Rhythm

Marina Granell Díaz

## Resumen

Documento de Diseño de Gamification Quest: Rhythm, un juego integrado en una aplicación de gamificación de cursos académicos.



## Contenido

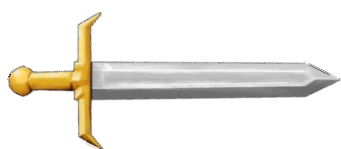
Resumen .....	1
Tabla de Imágenes .....	2
1. Concepto .....	3
2. Gamificación .....	3
3. Gameplay .....	6
4. Mecánicas de Juego .....	6
5. Personajes .....	8
6. Entorno .....	8
7. Comportamientos Inteligentes .....	10
8. Interfaz de Usuario .....	12
9. Controles .....	13
10. Flujo de Juego .....	14
11. Arte .....	15
12. Sonido .....	19

## Índice de Imágenes

Imagen 1. Pantalla de inicio de sesión y carga de la aplicación .....	4
Imagen 2. Pantallas exclusivas para usuarios profesores .....	5
Imagen 3. Menús del juego .....	5
Imagen 5. Interfaz de usuario .....	12
Imagen 6. Menú de pausa .....	13
Imagen 7. Representación de los controles del juego .....	13
Imagen 8. Nivel descrito en el flujo de juego .....	14
Imagen 9. Paleta de color de la interfaz .....	15
Imagen 10. Paleta de color del entorno .....	15
Imagen 11. Diferencias entre aspectos gráficos .....	19

## Índice de Tablas

Tabla 1. Instrumentos y su uso .....	6
Tabla 2. Efectos de los instrumentos .....	7
Tabla 3. Elementos interactivos y sus funciones .....	9
Tabla 4. Elementos decorativos del entorno .....	10
Tabla 5. NPCs y su comportamiento .....	10
Tabla 6. Comportamientos de elementos del juego .....	11
Tabla 7. Elementos 2D .....	15
Tabla 8. Elementos 3D del escenario .....	16
Tabla 9. Elementos 3D animados .....	17
Tabla 10. Efectos de sonido .....	19
Tabla 11. Pistas de música .....	20



## 1. Concepto

---

*Gamification Quest: Rhythm* es un juego integrado en una aplicación conectada a Moodle, cuyo objetivo es gamificar actividades académicas. El juego aprovecha la información recabada por la aplicación, detallada en la Sección 2, para adaptar la fuerza de los enemigos, del jugador y la dificultad de juego al desempeño académico del alumno.

En el juego, el personaje principal es un mago que controla su entorno y a ciertas criaturas por medio de la música. No hay una línea argumental que dicte el avance del juego, sino que éste dependerá del avance del alumno en sus estudios; por tanto, el juego se compone de niveles individuales, que se completarán cuando se cumplan los objetivos especificados (como recoger objetos) y el jugador llegue a la salida. Los niveles están formados por puzzles que el jugador deberá superar utilizando los instrumentos a su disposición. Completar cada nivel tiene una recompensa en la aplicación, que a su vez desbloqueará nuevos niveles de *Gamification Quest: Rhythm* a medida que el alumno avance.

## 2. Gamificación

---

El juego se integra en *Gamification Quest*, una aplicación desarrollada independientemente del juego. Esta aplicación está conectada al aula virtual de Moodle de un curso, recoge la información de los alumnos y las tareas y cuestionarios que completan, y la utiliza para desbloquear contenido.

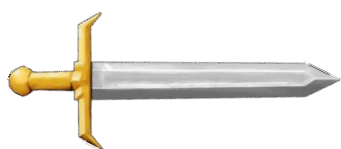
Para crear una cuenta en *Gamification Quest*, es necesario que el alumno tenga una clave y contraseña pertenecientes al aula virtual gamificada. Estos datos le servirán para activar su cuenta en la aplicación desde la pantalla representada en la Imagen 1. Tras conectarse por primera vez, la aplicación solicitará al jugador que introduzca un nombre de usuario por el que el resto de usuarios inscritos podrán reconocerle.

En concreto, *Gamification Quest* guarda información sobre:

- Las asignaturas en las que un alumno está inscrito
- Las tareas y cuestionarios abiertas en el momento de conexión para cada curso
- El estado de entrega de las tareas y cuestionarios del alumno (si ha sido completado, si el profesor los ha evaluado)

Además, la aplicación almacena información propia sobre cada jugador:

- Cada jugador tiene un nivel y estadísticas que aumentan con experiencia y puntos de habilidad, adquiridos tras completar tareas
- El juego realiza una clasificación de jugadores propia de cada jugador, de forma que todos los alumnos inscritos en alguna de las asignaturas en las que se encuentre el jugador en cuestión aparecerán en su clasificación, ordenados por nivel y méritos
- Cada jugador desbloquea nuevas misiones a su ritmo, gracias a las *Llaves de Mundo*, objeto obtenido cada vez que se completa una tarea. Cada *Llave de Mundo* puede canjearse por una nueva misión. El nivel tutorial está desbloqueado por defecto
- Para jugar misiones, el jugador debe gastar energía. La energía se recupera automáticamente con el paso del tiempo



- El jugador puede obtener familiares (criaturas que le ayudarán en las misiones) como premio por completar misiones. El primer familiar está desbloqueado por defecto
- También pueden obtenerse como recompensa *Orbes de Energía*, objetos de un solo uso que recuperan por completo la energía de un jugador

Toda esta información estará disponible mediante la aplicación para el jugador, que podrá ver el estado de compleción de sus entregas, los familiares y misiones que ha desbloqueado, su nivel, su puesto en la clasificación y más información en las pantallas representadas en la Imagen 3.

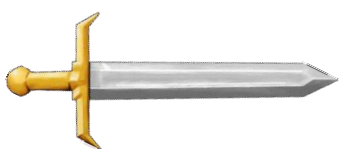
*Gamification Quest* también proporciona herramientas para que los profesores puedan configurar sus asignaturas desde una sección propia en la aplicación (representada en la Imagen 2):

- Para que un curso esté disponible para los alumnos, el profesor que lo administra debe activarlo primero desde la aplicación
- Los cursos también podrán desactivarse, resultando en la eliminación de todos los datos de tareas relacionadas con ese curso y la desaparición de ese curso en las cuentas de los alumnos inscritos
- El profesor puede marcar si una tarea es evaluable o no (si le va a poner nota en el aula virtual), y poner un peso que indique la importancia de cada tarea. Este peso, de 1 a 99 (por defecto 50), influirá en la cantidad de experiencia que el alumno reciba por completarla

El jugador accede a los niveles del juego mediante las misiones. Comenzar una misión transporta al jugador de la aplicación al nivel del juego enlazado a la misión seleccionada. Este nivel recibirá la información necesaria de la aplicación, como el tiempo que el jugador suele tardar en entregar tareas, y sus notas, para adaptar la fuerza de los enemigos y otros elementos a su experiencia.



Imagen 1. Pantalla de inicio de sesión y carga de la aplicación



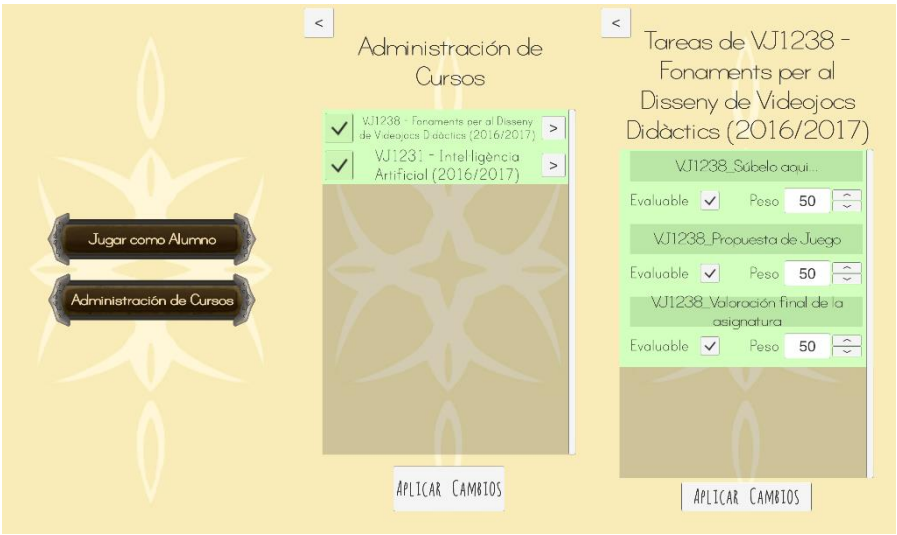
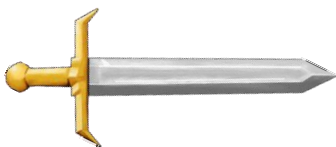


Imagen 2. Pantallas exclusivas para usuarios profesores



Imagen 3. Menús del juego





### 3. Gameplay

Debido a que el juego está integrado en la aplicación de gamificación, es necesario que el jugador complete tareas del aula virtual y obtenga *Llaves de Mundo* para desbloquear los niveles antes de jugarlos. El tutorial es una excepción, ya que siempre estará disponible.

En cada nivel, el jugador se encontrará en un mapa compuesto por múltiples obstáculos y enemigos, y estará acompañado por un familiar, elegido por él mismo antes de comenzar el nivel.

Los elementos del nivel, los enemigos y el familiar reaccionan ante la música. El jugador tiene a su disposición hasta tres instrumentos que tendrá que tocar para que la música siga sonando y entre los cuales podrá intercambiar para producir efectos variados. Al mismo tiempo, debe controlar el movimiento del personaje.

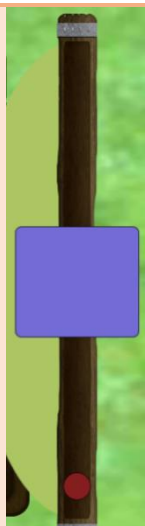
El objetivo dentro del nivel es encontrar un objeto, recogerlo, y encontrar un portal de salida que le permite regresar del nivel, de vuelta a la aplicación, todo ello sin dejar que la música pare o que los enemigos derroten al personaje o su criatura.

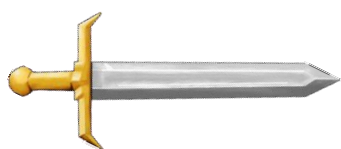
### 4. Mecánicas de Juego

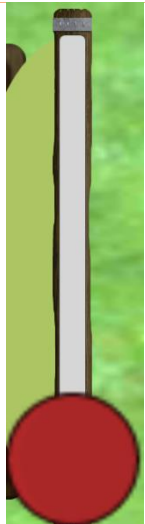
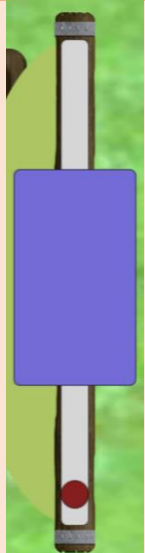
El juego se basa en el control simultáneo del movimiento del avatar y el instrumento. Para jugar el nivel, es necesario que el instrumento seleccionado no deje de sonar, es decir, se aceptan un máximo de fallos consecutivos dependiendo del instrumento en cuestión. Los fallos se traducen en daño para el jugador, y restan vida que podrá recuperar tocando correctamente el instrumento.

El elemento utilizado para tocar los instrumentos es una barra vertical con un indicador que se mueve arriba y abajo al compás de la música. Como se indica en la Tabla 1, hay una zona de acción con la que el jugador tiene que interactuar, aunque la forma de hacerlo depende del instrumento. En la Tabla 1 se describe la forma de tocar cada uno, junto con sus peculiaridades y efectos.

Tabla 1. Instrumentos y su uso

Instrumento		Cómo tocarlo	Fallos contabilizados
Mandolina		Pulsar sobre la barra cuando el indicador está situado en una pequeña zona de acción en el centro	<ul style="list-style-type: none"> <li>• No tocar la zona de acción antes de que el indicador salga de ella</li> <li>• Tocar cuando el indicador no está dentro de la zona de acción</li> </ul>



<b>Violín</b>		El jugador debe mantener el dedo encima del indicador móvil, mientras este se desplaza por la barra	<ul style="list-style-type: none"> <li>• Levantar el dedo del indicador</li> <li>• Alejarlo demasiado del indicador</li> </ul>
<b>Aulós</b>		La zona de acción ocupa el tercio central de la barra. El jugador debe mantener el dedo encima de la zona de acción mientras el indicador móvil esté situado dentro	<ul style="list-style-type: none"> <li>• No tocar el indicador antes de que se desplace demasiado del inicio de la zona de acción</li> <li>• Alejar demasiado el dedo del indicador cuando está dentro de la zona de acción</li> <li>• Soltar el indicador cuando todavía no ha llegado al final de la zona de acción</li> <li>• Tocar la barra cuando el indicador no está dentro de la zona de acción</li> </ul>

Los efectos que produce cada instrumento sobre los elementos interactivos (explicados en detalle en la Sección 6) están detallados en la Tabla 2, donde se detalla además el comportamiento del familiar tras seleccionarlos. Estos comportamientos se describen con mayor precisión en la Sección 7.

Tabla 2. Efectos de los instrumentos

<b>Instrumento</b>	<b>Efecto en el entorno</b>	<b>Efecto en el familiar</b>
<b>Mandolina</b>	<ul style="list-style-type: none"> <li>• Abre las fauces de las plantas carnívoras</li> <li>• Atrae a los enemigos estáticos</li> </ul>	Activa el modo acción: utiliza objetos en su posesión o interactúa con objetos del escenario dentro de su rango
<b>Violín</b>	<ul style="list-style-type: none"> <li>• Cambia la polaridad de los monolitos magnéticos</li> </ul>	Activa el modo de recolección: recoge automáticamente objetos que se encuentran en su rango
<b>Aulós</b>	<ul style="list-style-type: none"> <li>• Activa la polaridad de los monolitos magnéticos: el jugador será atraído o repelido por ellos</li> </ul>	Activa el modo defensa: se interpone en el camino de enemigos que quieran atacar al jugador e intercepta sus ataques



## 5. Personajes

---

En el juego aparecen tres tipos de personajes según su rol: el avatar, los familiares, y los enemigos. En esta sección se especifican sus funciones y la forma en la que interactúan entre sí.

Por una parte, el **avatar** es el nexo entre el jugador y el juego. Este personaje es controlado por el jugador, que dirige sus movimientos y controla los instrumentos que toca. El avatar está indefenso por sí mismo, ya que no tiene ninguna acción que le permita atacar, y tampoco puede interactuar con elementos del entorno (dado que está ocupado tocando el instrumento activo).

Son los **familiares** los que cumplen las funciones de las que el avatar carece, como el ataque y el uso de objetos. Aunque un jugador puede poseer múltiples familiares, como se indica en la Sección 2, sólo puede llevar uno consigo cuando entre en un nivel. Cada familiar tiene características distintas; los elementos que varían de uno a otro son, entre otros, *stats* como el ataque, defensa y vida, y la velocidad de movimiento cuando realiza acciones. El jugador controla al familiar elegido mediante la música, de forma que, dependiendo del instrumento, realizará acciones distintas (ver Sección 7 para una explicación detallada de la relación). El familiar puede recoger objetos, usarlos, interactuar con elementos del entorno y defender al jugador de enemigos hostiles, y siempre seguirá al jugador cuando no esté realizando ninguna de estas acciones.

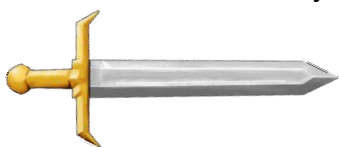
Por último, los **enemigos** son NPCs (*Not Playable Characters*) que pueden causar daño al avatar. Se encuentran situados en distintos puntos del nivel, y reaccionan ante la música y la posición del jugador. Según su comportamiento, hay dos tipos de enemigos: los ofensivos, y los pasivos.

- Los enemigos ofensivos patrullan dentro de una zona delimitada, sin excederla, esperando a que el jugador esté a su alcance. En cuanto el jugador está suficientemente cerca, le perseguirá, saliendo de la zona delimitada de ser necesario. El enemigo ofensivo daña al jugador al hacer contacto físico cada cierto tiempo. Para defenderse, el jugador tiene que activar el modo defensa del familiar con el aulós (ver Tabla 2). Mientras el jugador acierte tocando el instrumento, la criatura se defenderá con éxito, pero cada fallo restará vida de la barra de salud.
- Los enemigos pasivos están situados en un punto concreto del nivel y no se mueven ni persiguen al jugador a no ser que esté tocando la mandolina. La mandolina les atrae, y hace que, de haber contacto visual entre el enemigo pasivo y el jugador, persigan al avatar. Si el enemigo encuentra obstáculos entre su posición y la del jugador, le ignorará hasta que haya un camino libre en línea recta entre ellos. Cuando el jugador deja de tocar la mandolina, el enemigo pasivo vuelve a su sitio.

## 6. Entorno

---

El entorno constituye una gran parte de la jugabilidad, ya que sus elementos conforman los puzzles espaciales que el jugador deberá superar para completar el nivel. Los elementos interactivos están distribuidos de forma que reaccionen múltiples elementos de distinta forma cuando el jugador toque un instrumento, para crear conflictos y proporcionar,



preferiblemente, una sola solución para acceder a cada zona. El jugador deberá observar el entorno para encontrar la combinación correcta de instrumentos que le permita avanzar.

Tabla 3. Elementos interactivos y sus funciones

Elemento interactivo	Función	Cambio de comportamiento
<b>Planta carnívora</b>	Bloquea el camino. Ataca cuando el jugador se acerca.	Cuando el jugador toca la mandolina, abre las fauces y permite el paso.
<b>Monolito magnético</b>	Tiene una polaridad positiva o negativa que afecta al jugador cuando se activa su polaridad, atrayéndole o repeliéndole. Puede hacer que el jugador supere obstáculos y se mueva rápidamente.	Cuando el jugador toca el violín, invierte la polaridad de todos los monolitos del nivel.
<b>Tabla de madera</b>	Puede colocarse en las fauces de las plantas carnívoras para mantenerlas abiertas cuando intenten cerrarse.	Cuando el jugador toca el violín, el familiar puede recoger objetos. Cuando el jugador toca la mandolina y si tiene la tabla de madera cogida, puede utilizarla colocarla sobre una planta abierta.
<b>Objetivo del nivel</b>	Objetivo que el jugador ha de recuperar antes de dirigirse a la salida. No tienen una función dentro del nivel, pero pueden desbloquear instrumentos o criaturas y recuperar energía. Estarán disponibles en la aplicación descrita en la Sección 2.	Cuando el jugador toca el violín, el familiar puede recogerlo. Una vez recogido, desbloquea la salida
<b>Portal de salida</b>	Permite terminar el nivel una vez completado el objetivo.	Cuando el jugador lo toca y si ha completado el objetivo, termina el nivel con éxito, devolviendo al jugador a la aplicación descrita en la Sección 2, donde recibirá las recompensas especificadas.

El movimiento está limitado a un plano 2D (X, Z); el jugador no puede saltar ni existen elevaciones a las que pueda acceder. Además, los elementos no interactivos limitan el avance del jugador, definiendo la topología del nivel. Algunos elementos permiten el paso a través si el jugador se ve atraído por el monolito magnético.

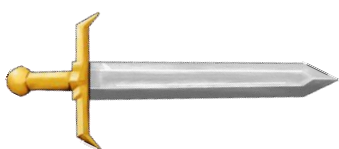


Tabla 4. Elementos decorativos del entorno

Elemento decorativo	Descripción	Bloquea el camino
Árbol (I, II, III)	Variedades de árboles. Múltiples texturas disponibles.	Sí
Roca (I, II)	Variedades de rocas. Múltiples texturas disponibles.	Sí
Arroyo	Animado con un <i>shader</i> de agua.	Sí, superable con el monolito magnético
Vegetación (I, II, III)	Distintos tipos de flores y plantas.	No

El escenario experimenta cambios visuales dependiendo del instrumento que el jugador utilice en el momento dado. En la Sección 11 se detalla el ambiente que se pretende reflejar con cada instrumento. Estos cambios tienen como objetivo la inmersión del jugador en las sensaciones que provoca la música que produce cada instrumento. Los objetos cambiarán cualidades de su material para dar la sensación de variación buscada, además de producirse modificaciones en la intensidad y color de la iluminación del entorno.

El *skybox* del nivel muestra una escena de día sencilla; por lo tanto, la iluminación refleja el momento, simulando una mañana soleada. La luz se atenuará o cambiará de color dependiendo de las modificaciones causadas por los instrumentos.

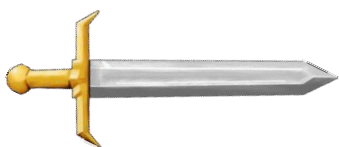
## 7. Comportamientos Inteligentes

Ciertos elementos y personajes del juego no controlados por el jugador muestran comportamientos inteligentes basándose en su entorno y el estado de la partida. En la Tabla 5 se detallan los comportamientos de personajes móviles, mientras que en la Tabla 6 se especifican conductas directamente relacionadas con el instrumento que el jugador está utilizando en un momento dado.

Tabla 5. NPCs y su comportamiento

NPC	Movimiento	Comportamiento
Enemigo hostil	<ul style="list-style-type: none"> <li>Patrulla cuando el avatar no está en rango<sup>1</sup></li> <li>Persigue cuando está en rango</li> </ul>	<ul style="list-style-type: none"> <li>Ataca siempre pasado un tiempo de inactividad al ritmo del instrumento</li> <li>Si la criatura está en modo defensa, se dirige a la criatura en vez de al jugador</li> </ul>
Enemigo estático	<ul style="list-style-type: none"> <li>Permanece quieto si no está en rango</li> <li>Persigue cuando está en rango</li> </ul>	<ul style="list-style-type: none"> <li>Embiste cuando el avatar está inmediatamente delante</li> </ul>

<sup>1</sup> Dentro de un radio alrededor del elemento en cuestión, que marca la distancia máxima de detección

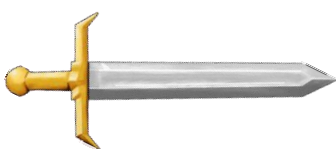


<b>Familiar</b>	<ul style="list-style-type: none"> <li>• Sigue al avatar</li> <li>• Si tiene que recoger objetos o realizar acciones, acude al objetivo</li> </ul>	<ul style="list-style-type: none"> <li>• Permanece con el jugador siempre y cuando no tenga que recoger un objeto o interactuar con un elemento</li> </ul>
-----------------	--	--

El familiar incorpora comportamientos relacionados con el movimiento más complejos que los de los enemigos, ya que el jugador tendrá constantemente en el rango de visión su comportamiento. Para simular un comportamiento inteligente, se incorporarán *steering behaviors*, comportamientos de direccionamiento del movimiento que harán que el familiar se mueva siguiendo trayectorias menos predecibles.

Tabla 6. Comportamientos de elementos del juego

<b>Elemento</b>	<b>Instrumento</b>	<b>Acción</b>
<b>Planta carnívora</b>	Mandolina	Mantiene la planta abierta
<b>Monolito magnético</b>	Aulós	Activa la polaridad del monolito
<b>Monolito magnético</b>	Violín	Invierte la polaridad de los monolitos (de positiva a negativa y viceversa)
<b>Criatura</b>	Mandolina	Entra en modo interacción
<b>Criatura</b>	Violín	Entra en modo recolección
<b>Criatura</b>	Aulós	Entra en modo defensa



## 8. Interfaz de Usuario

La interfaz está adaptada para ser controlada por medios táctiles en un dispositivo móvil, como se indica en detalle en la Sección 9. La pantalla del nivel se compone de los siguientes elementos, representados en la Imagen 4:

- Joystick virtual: Elemento interactivo constituido por una zona de movimiento y el controlador móvil.
- Mapa: Se agranda al pulsar encima. Indica la situación del objetivo y la salida en una silueta simplificada de la estructura del nivel, sin la situación de enemigos y otros elementos.
- Indicador de vida: Barra indicadora en la que aparece representada la vida del jugador. Se modifica cuando el jugador recibe daño de un enemigo y cuando restablece salud mediante un objeto.
- Barra de acción: Elemento interactivo para tocar los instrumentos. Su aspecto y funcionamiento se ve modificado dependiendo del instrumento seleccionado.
- Botones de selección de instrumentos: Cada botón representa un instrumento diferente. El botón del instrumento seleccionado aparece desactivado.
- Botón de pausa: Cerca de la barra de acción, para que el juego pueda pausarse rápidamente de ser necesario.



Imagen 4. Interfaz de usuario

Al pulsar el botón de pausa, el juego despliega un menú con opciones que permite ajustar parámetros y reanudar el juego, representado en la Imagen 5.

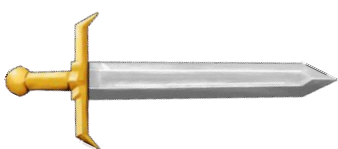






Imagen 5. Menú de pausa

## 9. Controles

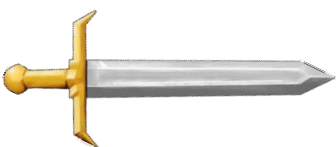
Control plenamente táctil adaptado a un dispositivo móvil en posición horizontal.

- Mano derecha controla el instrumento (cambiar de instrumento, ritmo).
- Mano izquierda controla el movimiento con un joystick virtual.
- Interacción con menús mediante entrada táctil (botón de pausa, selección).

El control es complejo para el jugador debido a que debe coordinar una acción rítmica con el movimiento del personaje, además de analizar la situación para avanzar en el nivel.



Imagen 6. Representación de los controles del juego



## 10. Flujo de Juego

El jugador lee la descripción del nivel y sus recompensas. Debe recoger un orbe de energía y encontrar una salida. Acepta la misión, selecciona el familiar que va a acompañarle, y el nivel comienza.

Se encuentra en el punto de inicio, y pulsa en la barra de acción para empezar a tocar la mandolina. La música empieza a sonar, el jugador mantiene el ritmo mientras contempla la situación. El mapa indica que el objetivo se encuentra en dirección norte, donde hay una planta carnívora. La planta carnívora está abierta gracias a la música de la mandolina, así que decide pasar a través.

Tras cruzar, el jugador se encuentra en una nueva sala con un enemigo pasivo, que se dirige hacia él al escuchar el sonido de la mandolina. Cambia de instrumento al violín. El enemigo se queda quieto, y el familiar recoge una tabla de madera dentro de su rango, después de entrar en modo recolección. El enemigo estático en frente está bloqueando el pasillo por el que se encuentra el objetivo, así que el jugador se coloca en la dirección en la que quiere que vaya y toca la mandolina de nuevo. De esta forma, el enemigo corre hacia el jugador en la dirección deseada. Vuelve a tocar el violín para que los enemigos no se sigan acercando.

El pasillo queda desbloqueado por el enemigo estático, pero una planta carnívora se encuentra en él, cerrada mientras toca el violín. El jugador vuelve a tocar la mandolina por un corto tiempo, lo suficiente como para que los enemigos no se acerquen demasiado, pero el familiar entra en modo interacción y coloque la tabla de madera sobre las fauces abiertas de la planta carnívora, manteniéndola así después de cambiar de nuevo al violín.

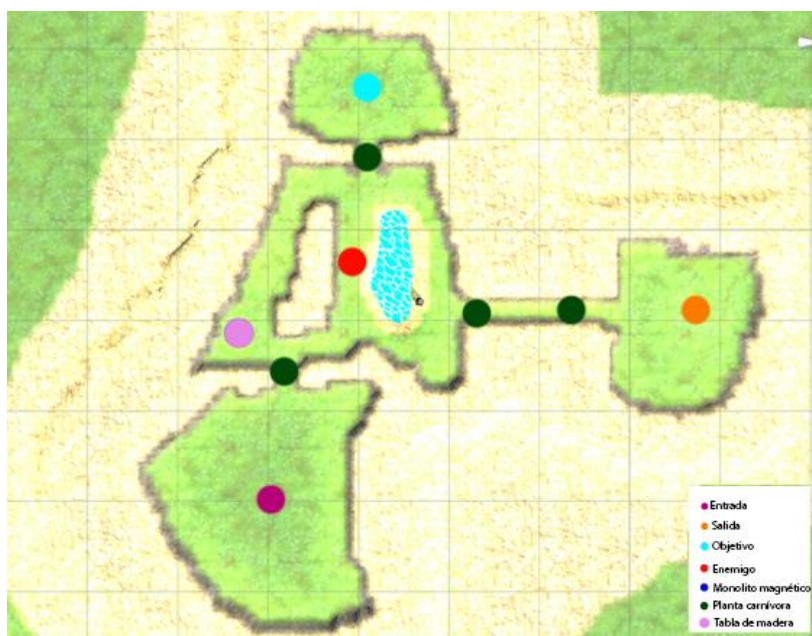
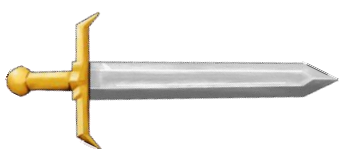


Imagen 7. Nivel descrito en el flujo de juego

Ahora el jugador es libre de pasar por la tabla, recoger el objetivo en la nueva estancia, y volver en dirección a la salida, que aparece en el mapa después de completar el objetivo anterior. La salida se encuentra en dirección este desde la estancia de los enemigos estáticos. Por tanto, el jugador deberá guiar al enemigo con la mandolina para alejarlo del pasillo que conduce a la salida.



Sin embargo, el pasillo está bloqueado por plantas carnívoras, que sólo se abren cuando el jugador toca la mandolina, y no quedan tablas con las que mantenerlas abiertas. La única opción es correr hacia la salida tocando la mandolina, mientras los enemigos persiguen al jugador.

El jugador alcanza la salida, completando así el nivel con éxito.

## 11. Arte

### Tipos de Arte

Existen diferentes tipos de elementos gráficos en *Gamification Quest: Rhythm*. Por una parte, la interfaz se compone de imágenes 2D, mientras que los elementos pertenecientes al nivel son modelados 3D. Ambos tipos de elementos tienen un estilo artístico conexo en cuanto a contexto (elementos que evocan a la edad media y con aspecto mágico), pero utilizan paletas de colores distintas, como muestran la () y la (). La paleta de colores de la interfaz se compone de colores desaturados en comparación con las paletas del entorno 3D.

- Imágenes 2D: Elementos sin línea delimitadora, sombreado duro y aspecto plano.



Imagen 8. Paleta de color de la interfaz

- Modelados 3D: Modelados *low poly* (<1500 polígonos), dado que el juego está pensado para una plataforma móvil. Aspecto geométrico en elementos del escenario, redondeado para personajes. Texturas sencillas con degradados. Uso de un *toon shader* para obtener un sombreado liso y duro.



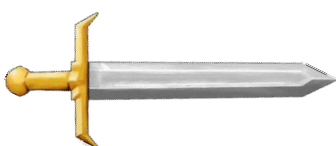
Imagen 9. Paleta de color del entorno

### Elementos 2D

Los elementos 2D son elementos de la interfaz gráfica que permiten la interacción con el juego y tienen carácter informativo.

Tabla 7. Elementos 2D

Nombre	Uso	Aspecto
<b>Botón de instrumento</b>	Activar instrumentos	Cuadrado, de madera
<b>Icono mandolina</b>	Dentro del botón de instrumento	Silueta de un solo color
<b>Icono violín</b>	Dentro del botón de instrumento	Silueta de un solo color
<b>Icono aulós</b>	Dentro del botón de instrumento	Silueta de un solo color
<b>Barra de acción</b>	Fondo que indica la zona de movimiento del indicador móvil del instrumento y su zona de acción	Madera con detalles metálicos

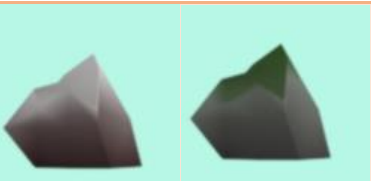
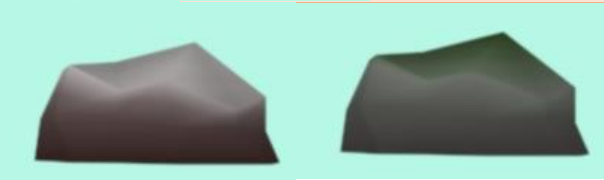




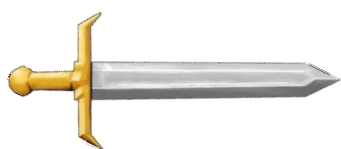
<b>Indicador</b>	Elemento móvil que se desplaza por la barra de acción	Pequeño, redondo, de un color que resalte sobre la barra de acción
<b>Joystick</b>	Elemento interactivo para mover el avatar	Semitransparente, redondo, con un motivo de los ejes de movimiento superpuesto
<b>Marcador mapa jugador</b>	Representa al jugador sobre el mapa	Estrella sencilla
<b>Marcador mapa objetivo</b>	Representa el siguiente objetivo en el mapa	Forma poligonal sencilla

## Elementos 3D

Los elementos 3D se dividen a su vez entre elementos estáticos o del escenario y elementos animados.

Tabla 8. Elementos 3D del escenario

Nombre	Aspecto
<b>Roca I</b>	
<b>Roca II</b>	
<b>Árbol I</b>	
<b>Árbol II</b>	








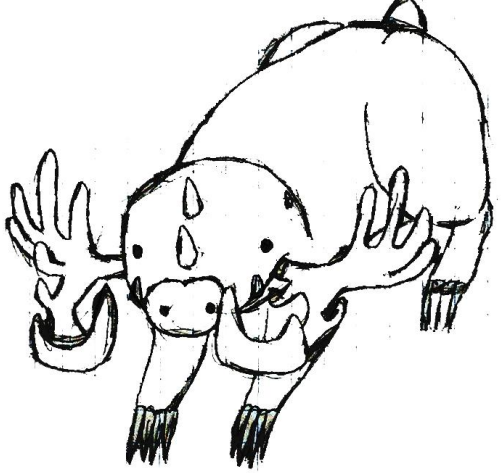

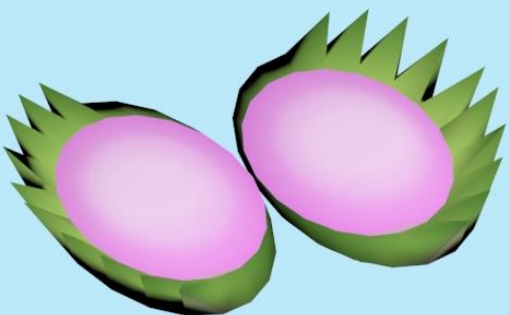
Árbol III		
Flores		
Hierbajos		
Portal de salida		

Tabla 9. Elementos 3D animados

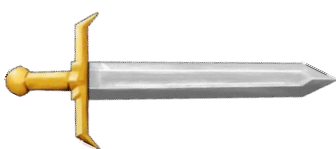
Nombre	Aspecto	Animaciones
Avatar		<ul style="list-style-type: none"> <li>• Estático</li> <li>• Caminar</li> <li>• Derrota</li> <li>• Victoria</li> <li>• Recibir daño</li> </ul>



<b>Familiar</b>		<ul style="list-style-type: none"> <li>• Movimiento</li> <li>• Atacar</li> <li>• Desfallecer</li> </ul>
<b>Enemigo estático</b>		<ul style="list-style-type: none"> <li>• Estático</li> <li>• Correr</li> <li>• Embestir</li> </ul>
<b>Enemigo hostil</b>		<ul style="list-style-type: none"> <li>• Movimiento</li> <li>• Atacar</li> <li>• Desfallecer</li> </ul>
<b>Planta carnívora</b>		<ul style="list-style-type: none"> <li>• Abrir/Cerrar boca</li> </ul>

## Cambios en el Aspecto Gráfico

El aspecto gráfico del juego se ve modificado por las acciones del jugador. Existen tres posibles aspectos gráficos para el nivel y todos sus elementos: el de la mandolina, más iluminado y soleado; el del violín, idílico y azulado; y el del aulós, crepuscular y más oscuro. La Imagen 10 muestra una comparación entre los tres aspectos.





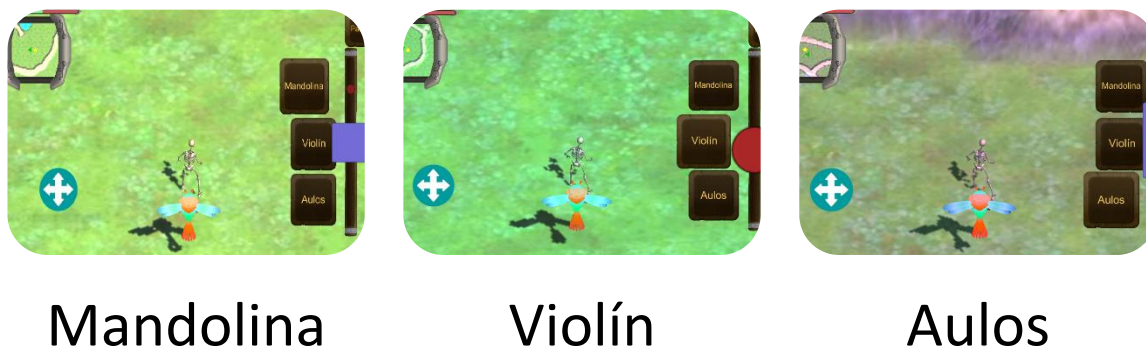


Imagen 10. Diferencias entre aspectos gráficos

## 12. Sonido

La música del juego es un factor relevante para la jugabilidad y es aspecto gráfico del mismo. Los instrumentos que el jugador puede tocar (aulós, mandolina y violín) son las voces principales de los tres temas que se alternan durante el nivel, dependiendo de qué instrumento esté sonando. Así mismo, aunque todos juntos forman una sola melodía, por separado también constituyen temas completos, que transmiten distintas sensaciones (más animado, tenso, tranquilo). Estas sensaciones se reflejan en el entorno mediante cambios de iluminación descritos en la Sección 11.

Los instrumentos han sido elegidos cuidadosamente entre aquellos existentes en una época medieval de fantasía. Cada uno representa un tipo de instrumento: la mandolina para la cuerda pulsada, el violín para la cuerda frotada, y el aulós para el viento madera.

### Efectos de Sonido

Con tal de aumentar la respuesta del juego a las acciones del jugador, es necesario el uso de efectos de sonido. En la Tabla 10 se enumeran los efectos de sonido previstos para un correcto *feedback* auditivo.

Tabla 10. Efectos de sonido

Nombre	Descripción
<b>Botón</b>	Sonido sordo y seco, para pulsar los botones de la interfaz
<b>Pulsar instrumento</b>	Sonido discreto para indicar el uso de la barra de acción
<b>Mantener instrumento</b>	Sonido discreto para indicar el uso de la barra de acción
<b>Fallo tocando un instrumento</b>	Producido por una interacción errónea con la barra de acción
<b>Fallo sin tocar un instrumento</b>	Producido cuando no se interactúa con la barra de acción durante un compás
<b>Ataque amigo</b>	Sonido de ataque mágico producido por el ataque del familiar
<b>Ataque enemigo</b>	Sonido de mordisco producido por el enemigo hostil
<b>Recibir daño</b>	Quejido producido por el familiar al recibir un ataque
<b>Cumplir objetivo</b>	Melodía muy corta de éxito





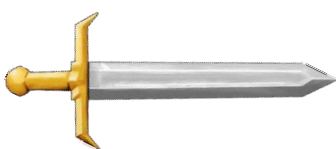
<b>Activar monolito</b>	Sonido de estática producido al activarse el campo magnético
<b>Desactivar monolito</b>	Sonido inverso al de activación
<b>Pisadas del avatar</b>	Pisadas livianas sobre tierra
<b>Pisadas del enemigo estático</b>	Pisadas pesadas sobre tierra, de un animal corpulento
<b>Embestida del enemigo estático</b>	Golpe seco contra un objeto, producido cuando el enemigo estático impacta contra algo

## Música

En total, son necesarias las pistas de música especificadas en la Tabla 11. Todas las pistas deben tener la misma duración, ya que se reproducirán simultáneamente durante la partida. El volumen de cada pista cambiará en función de las acciones del jugador, de forma que sonarán sólo las pistas necesarias. La disonancia para el fallo es una excepción, ya que no contiene una melodía ni está prevista su reproducción prolongada.

Tabla 11. Pistas de música

<b>Nombre</b>	<b>Uso</b>	<b>Duración</b>
<b>Base musical</b>	Siempre suena durante el nivel. Contiene percusión y una base para la melodía que encaja con el resto de pistas.	1 minuto
<b>Mandolina</b>	Contiene la melodía de la mandolina. Sólo suena cuando el jugador usa la mandolina.	1 minuto
<b>Violín</b>	Contiene la melodía del violín. Sólo suena cuando el jugador usa el violín.	1 minuto
<b>Aulós</b>	Contiene la melodía del aulós. Sólo suena cuando el jugador usa el aulós.	1 minuto
<b>Disonancia para el fallo</b>	Nota grave sostenida. Cuando el jugador falla consecutivamente, disminuye el volumen de la base y suena la disonancia para indicar al jugador que puede perder la partida.	5 segundos



# Chapter 10

## Appendix B.

### Music and Sound Design

---

# Gamification Quest: Diseño de música y sonido

Por Víctor Ávila

La banda sonora de Gamification Quest ha sido compuesta para trasladar al jugador al mundo de fantasía con toques medievales que se plantea al juego. Para ello se decidió optar por unas melodías definidas desde un principio y ritmos bien marcados, dando dinamismo a las partidas.

Elegimos temas inspirados en obras populares medievales, tanto en sus armonías como en los procesos rítmicos, y por la estética del juego optamos por temas alegres a medios tiempos intercalando melodías menores, dando citas a los enemigos que vas a encontrarte en todo el camino.

Tenemos claro que el soporte del juego son teléfonos o tablets, es por ese motivo que adaptamos las sonoridades a estos dispositivos, pero cuidando al máximo la calidad de las bibliotecas de sonidos utilizando, llegando a rediseñar cada sonido utilizado para que casen perfectamente unos con otros, y no suene excesivamente artificial.

Cada sonido incidental utilizado ha sido diseñado específicamente para que el jugador conciba la acción del juego tanto visual como auditivamente: pasos, magia, etc.

En resumen, esta banda sonora ha sido un resultado del trabajo conjunto entre historia, jugabilidad y estética del juego, ofreciendo una experiencia tanto divertida como adictiva en el plano sonoro.

# Chapter 11

## Appendix C.

### Playtesting questionnaire

---

1 2 3 4 5

---

not at all ☐ ☐ ☐ ☐ ☐ completely

7. Is the illumination provided adequate?

Maqueu només un oval.

	1	2	3	4	5	
too dark	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	too bright

8. Are the illumination changes produced by changing instruments noticeable?

Maqueu només un oval.

	1	2	3	4	5	
not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	too noticeable

9. Is the art style attractive?

Maqueu només un oval.

	1	2	3	4	5	
not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	completely

10. Is the art style adequate to the genre?

Maqueu només un oval.

	1	2	3	4	5	
not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	completely

11. Do the graphics provide visual feedback to your actions?

Maqueu només un oval.

		1	2	3	4	5	
not enough feedback		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	graphics provide satisfying feedback

## Sound design & Music

12. Sound effects

Maqueu només un oval.

	1	2	3	4	5	
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable

13. Music

Maqueu només un oval.

	1	2	3	4	5	
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable

14. Is the music adequate to the game genre?

Maqueu només un oval.

	1	2	3	4	5	
not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	completely

15. Do the sound effects provide auditory feedback to your actions?

Maqueu només un oval.

	1	2	3	4	5	
not enough feedback	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sound effects provide satisfying feedback

## Input

## 16. Instrument playing and selection

Maqueu només un oval.

- ☐ too easy
- ☐ easy
- ☐ just right
- ☐ hard
- ☐ too hard

## 17. Movement control

Maqueu només un oval.

- ☐ too easy
- ☐ easy
- ☐ just right
- ☐ hard
- ☐ too hard

## General

**18. Did you find any bugs during the session?**

---

---

---

---

---

## 19. Overall grade

Maqueu només un oval.

0	1	2	3	4	5	6	7	8	9	10
										



# Chapter 12

## Appendix D.

### Project Exhibition Leaflet

---



Descarga la app y prueba el  
juego en tu móvil escaneando el  
código QR

SO mínima: Android 4.1 Jellybean

**Usuario: gamquest**  
**Contraseña: tfg2017**

Si quieres evaluar el juego  
después de probarlo, puedes  
escanear este código QR



**¡Muchas gracias por tu ayuda!**

Marina Granell Díaz  
Gamification Quest: Rhythm  
music as a game mechanic